Journal of Robotics and Control (JRC)
Volume 6, Issue 4, 2025
ISSN: 2715-5072, DOI: 10.18196/jrc.v6i4.25026

Optimizing Resource Allocation and Link Reliability in IoT–Fog–Cloud Networks Using Machine Learning and Multi-Objective Algorithms

M. Sri Lakshmi [1*], Palakeeti Kiran [2], S Suneetha [3], Buradagunta Swathi Sri [4], Srinivasa Rao Madala [5], Naresh Kumar Bhagavatham [6], Maloth Bhavsingh [7*]
[1] Department of Computer Science and Engineering, G. Pullaiah College of Engineering and Technology, Kurnool, India
[2] Department of Computer Science and Engineering (AI), Madanapalle Institute of Technology and Science, Madanapalle, India
[3] Department of Computer Science and Engineering, PACE Institute of Technology and Sciences, Ongole, India
[4] Department of Computer Science and Engineering, Koneru lakshamaih Education Foundation, Vaddeswaram, India
[5] Department of Artificial Intelligence and Data Science, Chaitanya Bharathi Institute of Technology(A), Gandipet, Hyderabad, India
[6] Department of Computer Science and Engineering, Vignana Bharathi Institute of Technology, Hyderabad, India.
[7] Department of Computer Science and Engineering, Ashoka Women's Engineering College, Kurnool, India
Email: [1] srilakshmicse@gpcet.ac.in, [2] kiranpalakeeti@gmail.com, [3] Suneetha_s@pace.ac.in, [4] swathimth1993@gmail.com, [5] mr.srinu13@gmail.com, [6] bhagavatham.nareshkumar@vbithyd.ac.in, [7] bhavsinghit@gmail.com
*Corresponding Author

***Abstract*—The internet of things (IoT) necessitates efficient real-time data transfer protocols to support its vast array of interconnected devices. This study presents an optimized framework for resource allocation and link reliability in IoT–fog–cloud networks by integrating an enhanced support vector machine (ESVM) for link stability prediction with a Communication and Energy Integration for latency improvement (CAELI) algorithm for multi-objective optimization. The proposed system improves the quality of service (QoS) by dynamically selecting energy-efficient, low-latency paths while accounting for network conditions and resource constraints. The ESVM leverages historical link characteristics to assess reliability, whereas CAELI minimizes communication delay and energy usage through adaptive optimization. The simulation results indicate that the model achieves consistent improvements across metrics such as link reliability, end-to-end delay, energy consumption, throughput, and packet delivery ratio (PDR), maintaining a PDR above 94%, which is particularly significant in real-time control systems where even minor packet loss can compromise operational integrity. A comparative analysis with existing baseline and recent optimization approaches demonstrated superior performance in both static and moderately dynamic network environments. However, the model's effectiveness may be influenced by factors such as network scale, node mobility, and the complexity of parameter tuning in CAELI, which can affect the convergence rate and computational efficiency. These limitations suggest the need for further validation in large-scale heterogeneous IoT deployments. The proposed framework underscores the viability of combining predictive modeling with multi-objective optimization to enhance responsiveness, energy efficiency, and reliability in distributed fog-assisted architectures for time-sensitive IoT applications.**

***Keywords*—IoT-Fog-Cloud Architecture; Link Stability Prediction; Enhanced Support Vector Machine; Communication and Energy Integration for Latency Improvement; Energy Efficiency; Real-Time Communication.**

I. INTRODUCTION

The proliferation of internet of things (IoT) technologies has led to a dramatic increase in the number of connected devices, resulting in high data generation across sectors such as healthcare, transportation, and energy [1], [2]. These applications often require stringent quality-of-service (QoS) parameters, including low latency, high reliability, and efficient resource utilization [3]. Traditional cloud-centric architectures cannot consistently meet these demands because of their inherent limitations, such as centralized processing, increased communication delay, and network congestion [4]. To address these issues, fog computing has emerged as a decentralized paradigm that brings computation, storage, and control closer to the network edge [5]. By deploying heterogeneous fog servers (FSs) near end devices, fog computing reduces latency, improves bandwidth efficiency, and enhances the responsiveness of delay-sensitive applications [6], [7].

Despite their advantages, fog-based systems face critical challenges in dynamic resource allocation, particularly under conditions of resource scarcity, heterogeneous device capabilities, and fluctuating network topologies [8]. Efficient provisioning in such environments must balance conflicting objectives, such as energy consumption, latency, link stability, and reliability [9]. Existing heuristic-based and rule-driven approaches often lack adaptability and fail to account for multi-objective trade-offs, thereby limiting their scalability and robustness in real-world deployments [10]. These limitations underscore the need for intelligent, predictive, and optimization-driven methods that can operate under varying network conditions while maintaining the QoS and operational efficiency.

This study proposes a hybrid framework that integrates an enhanced support vector machine (ESVM) for predictive link

stability assessment and a multi-objective optimization algorithm, communication and energy integration for latency improvement (CAELI), to optimize routing and resource allocation in IoT–fog–cloud networks. By combining machine learning-based reliability estimation with energy-aware and latency-optimized decision-making, the framework aims to improve communication efficiency, path stability, and overall QoS in real-time IoT scenarios.

The contributions of this study are as follows.

- Development of an enhanced SVM-based model to predict link stability under variable network conditions and improve reliability in dynamic environments.
- Integration of a multi-objective CAELI algorithm to identify Pareto-optimal paths by balancing communication delay and energy consumption.
- A comprehensive evaluation of the proposed model against baseline and state-of-the-art approaches demonstrated an improved PDR, reduced delay, and higher energy efficiency.
- Identification and discussion of model limitations, including convergence complexity and environmental sensitivity, to guide future improvements.

The remainder of this paper is structured as follows: section II reviews the related work. Section III describes the proposed ESVM–CAELI framework. Section IV discusses the experimental setup, followed by the results and analysis in section V. Section VI concludes the paper with insights into the limitations and future directions.

II. EASE OF USE

There is an increased interest in resource optimization in fog computing, which has given rise to more diverse methods for enhancing resource management. In this study [11], the author proposed a multi-objective dynamic resource allocation approach using GA and RL to address workload issues. Instead, they depend on constant surveillance of huge traffic and loads within the network, but may be challenged in managing fluctuating power demands and maintaining stability in IoT-fog integration.

In a recent study [12], the authors presented an energy-focused load distribution strategy based on the energy capabilities of fog devices, which improves energy conservation. However, this method lacks the ability to improve the response time and reliability, which are important in IoT applications.

In this work [13], the author introduced a blockchain-based deep RL scheme for energy-adaptive job scheduling by exploiting an Asynchronous actor-critic agent model. Although efficient for task offloading and scheduling, the framework does not address important issues such as link reliability and holistic energy management in fog systems. In a study [14], the author proposed an ant mating optimization technique for tasks with the objective of addressing energy consumption and system lifetime; however, it is not flexible enough to provide a real-time response for IoT.

In a previous study [15], the authors presented MinRes and MinEng strategies to decrease response time and energy utilization. Although these methods are targeted to be efficient, they offer no comprehensive solution for addressing the needs of IoT-fog ecosystems. In SDN-assisted edge networks, the author of a research proposal [16] implemented deep Q-learning for dynamic job scheduling owing to its energy-constrained nature. Although this approach improves energy utilization efficiently, it is not very efficient in dealing with the complexity and heterogeneity that are characteristic of IoT environments.

Other research contributions include the research presented by Author [17], who focused on using machine learning models to enhance power and service placement, and Author [18], who developed power efficiency in fog computing. Such attempts help save energy, but they may not sufficiently meet the path reliability and link stability of the IoT-fog network. The realization of energy-efficient routing for a low-energy network was the topic of discussion by the authors [19], and [20] reviewed the use of machine learning in IoT-fog computing. Both studies focused on energy consumption but may not be sufficient to support the high-speed response rates of IoT systems.

The authors of these studies [21], [22] have implemented new approaches for energy-conscious design, and multi-facet scheduling and resource management have been enhanced. However, attaining stability and energy optimization, along with real-time IoT applications, remains a concern. Finally, in [23], the author discussed energy-efficient offloading policies to enhance the overall operational time in IoV systems. While useful, these techniques can seem deficient in terms of capturing the dynamic IoT-fog environment to accommodate inherent scalability and flexibility.

A. Research Gap

Realizing and understanding some limitations involved in the earlier studies, this research proposes integrating a new technique that comprises a new support vector machine (SVM) and additional multiple objective CAELI. The proposed method focuses on addressing the combined problem of link stability and energy consumption minimization in IoT-fog computing. While prior methods mainly addressed energy consumption and resource management, our fully integrated approach guarantees dependable and continuous connectivity, which is the basis of IoT-fog environments' fast-paced flows.

III. USING THE TEMPLATE

The proposed architecture was designed to enhance the performance of IoT–fog–cloud networks by integrating predictive modeling and multi-objective optimization for efficient resource allocation and communication path selection. The system comprises three hierarchical layers: the IoT layer, consisting of heterogeneous sensing and actuator devices; the fog layer, composed of intermediate fog servers responsible for edge-level processing and short-term data handling; and the cloud layer, which provides centralized storage, global analytics, and long-term orchestration.

At the core of the framework is dual-module methodology. First, the enhanced support vector machine

(ESVM) model predicts the reliability of each communication link by analyzing historical link performance metrics, such as the signal-to-noise ratio, packet loss rate, and delay trends. This classification enables the early elimination of unstable paths. Second, the communication and energy integration for latency improvement (CAELI) algorithm performs multi-objective optimization by selecting optimal communication paths that minimize both energy consumption and end-to-end latency.

As shown in Fig. 1, the workflow proceeds in sequential stages, beginning with data acquisition from the IoT nodes and link metric collection. This is followed by ESVM-based stability scoring, after which CAELI executes its optimization routine to determine the Pareto-optimal routing paths. The final routing table generated is deployed across the fog layer for real-time decision making. The IoT-fog-cloud computing network architecture in this study aims to establish an efficient communication model for smart applications through the three layers of IoT, fog, and cloud.

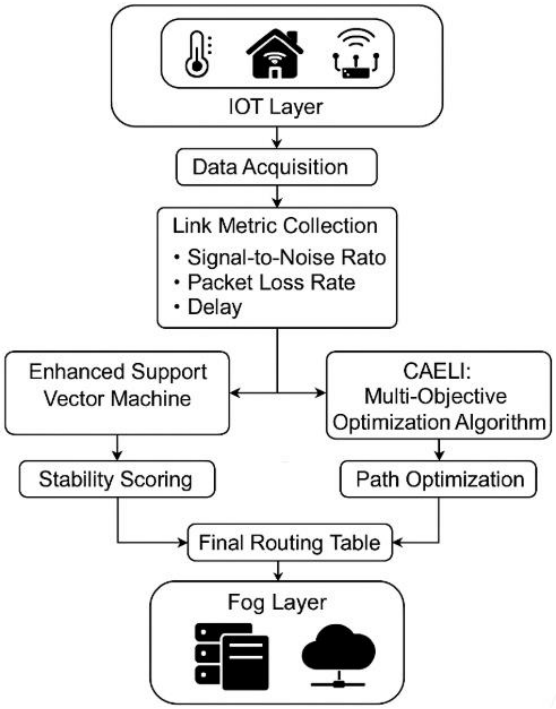

Fig. 1. Proposed methodology flow diagram

In this architecture, as shown in Fig. 2, the fog layer is very central in processing the requests from IoT devices and in ensuring that data transfer is conducted through energy-efficient and reliable channels. To this end, an enhanced support vector machine (ESVM) is used to estimate link stability and CAELI to choose the best paths for communication. The model fulfills essential quality of service (QOS) characteristics by reducing energy and increasing route stability. IoT flows in the model are described as $F(D_f, \delta_f)$, where D_f is the loT device producing the flow and δ_f>0 is the flow delay. It adheres to transmission delay constraints that constrain communication between the various layers within the system; link stability is obtained by relying on the reliability of the link and environmental aspects. The stability of links and energy levels are used as the major QoS parameters, which define the availability of reliable paths. Link stability prediction using machine learning and multi-objective optimization for path selection are generally incorporated in the proposed architecture to improve the communication paths and overall performance of the network. The integration of ESVM and CAELI enhances the transfer of data from IoT devices to fog nodes and the cloud server and provides a sustainable solution for complicated scenarios in IoT, where the real-time processing of data is essential and energy utilization is significantly important.

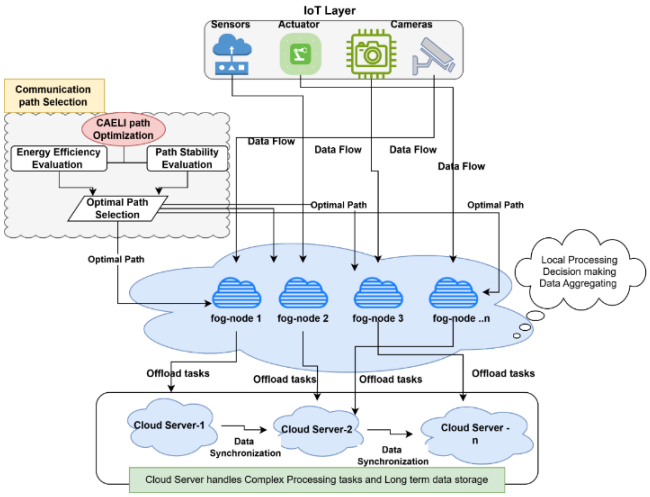

Fig. 2. IoT-fog network architecture

A. *Network Architecture*

The proposed IoT-fog-cloud computing network combines the IoT, fog, and cloud layers to design a communication system, as depicted in Fig. 2. The fog layer plays an indispensable role in handling requests originating from IoT devices and establishing data transmission through energy-efficient and stable pathways. During this process, an ESVM algorithm is used to predict link stability, and CAELI subsequently chooses the best communication paths in the multi-objective optimization of link stability and energy level [24]. These selected paths satisfy the QoS demands by focusing on path stability and relatively low energy expenditure. An IoT flow $F(D_f, \delta_f)$ is defined, where D_f is the IoT device generating the flow and $\delta_f > 0$ is the flow's delay. Transmission delays are enforced for each flow f.

1) *Link Stability*

Link stability (S) and energy levels (δ) are key QoS attributes [25], [26]. The network G can experience link failures due to external uncontrollable events ($e_s \in E$). Link stability is assumed to be independent, and the reliability level $R(l_i \mid e_s)$ of a link l_i is calculated as shown in (1):

$$R(l_i \mid e_s) - S_{t_i}(l_i \mid E) - 1 - \sum_{e_i \in E} \left(1 - S(l_i \mid e_s)\right) \quad (1)$$

Where $R(l_i \mid e_s)$ is the reliability of link l_i given event e_s, $S_{t_i}(l_i \mid E)$ represents the link's overall stability considering all events [27], [28] in E, and $S(l_i \mid e_s)$ is the stability level for each event e_s. The stable path $S(p_x)$ from source node D to fog server Fs_n for a flow, considering all events in E, is expressed in (2):

$$S(p_x(D, Fs_n) \mid E) - \sum_{l_i \in p_x \in P' \mid E} \left[S_l(D, Fs_n) \times S_{t_i}(l_i \mid E)\right] \quad (2)$$

Here, $S(p_x(D, Fs_n) \mid E)$ indicates the stability of path p_x from D to $F_{s_n} S_T$ is the relisbility indicator for a link in path p_x, and $S_{t_i}(l_i \mid E)$ denotes the stability level of link l_i under events E.

Path and link status correlation [29], represented by $\mathcal{C}(p_x)$ and $\mathcal{C}(l_i)$, respectively, involves data on energy and stability levels [30], [31]. The path status $\mathcal{C}(p_z)$ also includes the QDS matrix [32]. From these correlations, two main conclusions can be drawn.

- The status of each path $\mathcal{C}(p_x)$ and its constituent links $l_j \in p_x$ influences the fink's condition.
- The condition of each link $\mathcal{C}(l_i)$ affects the path status $l_i \in p_x$.

These relationships are defined in (3).

$$\mathcal{C}(l_i) = \zeta\left(\mathcal{C}(p_{x1}), \dots, \mathcal{C}\left(p_{pjj}\right)\right), \forall l_i \in p_{x,} x = 1,2, \dots, j \text{ and } \mathcal{C}(p_x) = \zeta\left(\mathcal{C}_{L_2(1)}, \dots, \mathcal{C}_{L_0 \| p_x \|}\right) \quad (3)$$

Where ζ and ζ are optimization functions. Direct optimization of ς and ζ is not feasible, so a multi-objective [33], [34] enhanced whale optimization technique is applied to derive their structure [35], [36].

2) Average Energy Consumption

Average energy consumption [37] is defined as the normalized power consumption of various fog [38] and IoT nodes [39] over time T. This is expressed as (4):

$$A_E(t) = \frac{\sum_{d_i \in D} \int_{t=S_E+(T_t)}^{S_F(t_t)} P_{d_i}(t) dt}{|A_I t| \sum_{d_i \in D} P_{d_i}^{\max} \times (t_{i+1} - t_i)} \quad (4)$$

Where $P_{d_i}^{\max}$ denotes the maximum power capacity of device d_i and $P_{d_i}(t)$ represents the energy function of node d_i during time T.

B. Objective Function

Each stream f in the network is associated with QoS attributes [40], specifically route stability [41] and energy consumption [42]. The objectives of this study are as follows:

1) Path Stability (maximization)

The goal is to identify a route [43] with the highest link reliability [44], ensuring maximum path stability [45]. The objective function for maximizing path reliability is expressed as follows in (5):

$$\max\left(\min\left(\sum_{p_x \in P' \mid E} S_{p_x} S_l(D, Fs_n)\right)\right) \quad (5)$$

Where S_{p_x} represents the stability of path p_x and S_I is the reliability indicator for the path from the source node D to the fog node Fs_n.

2) Energy Consumption (Minimize)

The objective is to find a path that minimizes energy usage [46], calculated as the average energy consumption across all links constituting the path $p_z(D, Fs_n)$, where Fs_n denotes the fog nodes. This is represented by (6).

$$\min \sum_{p_x \in P} A_E(t) \quad (6)$$

Here, $A_E(t)$ signifies the average energy consumption over time for the path p_x.

C. Support Vector Machine method in Link Stability Prediction: ESVM

This work also presents the developed ESVM model that employs machine learning for the fine prediction of network link stability [47]. Connection failure has often been the focus, and the ESVM is trained with past data for the corresponding environments in the observed networks. This is why improvements include turning on a soft margin to allow students to include positive, negative, and noncommittal values. The adaptation of the γ (gamma) parameter is an important modification that allows for high classification rates, especially in high-dimensional data spaces [48]. This refined approach to hyperplane configuration will make the estimation of link stability more precise and accurate.

In this area, the ESVM model has been shown to outperform other methodologies, including radial basis function (RBF) kernels, in predicting network link stability and classifications [49]. This is crucial for reliable communication between IoT and fog because it allows for the prediction and prevention of link failure. The generic form of the ESVM classifier function is presented in Eq (7).

$$f(z) = \sum_{i=1}^{n} \xi_i \, \mathrm{K}(z, z') \quad (7)$$

Where $f(z)$ represents the classifier function for the hyperplane, ξ_i denotes the weights assigned to training data points, $K(z, z')$ is the Kemel function, and z and z' are the feature vectors.

Kernel function based on Euclidean distance: The kernel function $K(z, z')$, based on the Euclidean distance, is expressed as:

$$K(z, z') - \exp\left[-\rho \| z - z' \|^2\right] \quad (8)$$

Where $\| z - z' \|^2$ represents the squared Euclidean distance between the feature vectors.

Unbound parameter in RBF classifier: For the RBF classifier [50], [51], the kernel function incorporating the unbound parameter α is shown in (9). The RBF classifier function is then defined as (10)

$$K(z, z') - \exp\left[-\frac{\| z - z' \|^2}{2\alpha^2}\right] \quad (9)$$

$$f(z) = \sum_{i=1}^{n} \xi_i \exp\left[-\frac{\| z - z' \|^2}{2\alpha^2}\right] \quad (1)$$

Where ξ_i represents the weights of the training data points. $\|z - z_i'\|_2^2$ is the squared Euclidean distance between the feature vectors z and z_i'.

D. CAELI: Multi-objective Optimization Algorithm

The Communication and Energy Integration for Latency Improvement (CAELI) algorithm is a multi-objective optimization model [50], [51] designed to improve route selection in IoT–fog–cloud networks. It addresses two key performance metrics—energy consumption and end-to-end latency—using a bio-inspired metaheuristic approach [52]. CAELI operates on a population of potential routing paths and evolves optimal solutions through a series of adaptive phases. This subsection details the operational phases and core functions of CAELI.

1) Objective Functions

The total cost function is defined as a weighted combination of energy and latency [53], as shown in (11):

$$f_{\text{total}} = \alpha \cdot f_{\text{energy}} + \beta \cdot f_{\text{latency}} \tag{11}$$

Where f_{energy} is the total energy cost along a selected path, f_{latency} is the cumulative delay from source to destination, α and β are weight coefficients representing application priorities ($\alpha + \beta = 1$). The objective is to minimize f_{total} while maintaining link reliability as determined by ESVM.

2) Phase I: Prey Searching

In this initialization phase, CAELI identifies a diverse population of potential routing paths using a probabilistic path discovery mechanism [54]. Each candidate path P_i is constructed based on neighborhood exploration and historical link stability scores provided by the ESVM module [55]. The goal is to ensure sufficient diversity in early search stages to avoid local minima.

3) Phase II: Prey Encircling

This phase enhances exploitation by drawing candidate paths towards the best-performing solutions using adaptive position updates, as shown in (12):

$$\vec{P}_i^{t+1} = \vec{P}_i^t + r_1 \cdot \left(\vec{P}_{\text{best}} - \vec{P}_i^t\right) \tag{12}$$

Where $\vec{P}_i^t$ is the current position (routing path) of solution i, $\vec{P}_{\text{best}}$ is the current best solution, and $r_1 \in [0,1]$ is a control parameter that adjusts exploration vs. exploitation.

4) Phase III: Bubble-Net Attacking

In this phase, the solutions are fine-tuned via spiral and contraction updates [56]. Inspired by the hunting behavior of humpback whales [57], the algorithm updates the paths using (13):

$$\vec{P}_i^{t+1} = \vec{P}_i^t \cdot \cos(2\pi l) + \vec{P}_{\text{best}} \cdot e^{bl} \tag{13}$$

Where l is a logarithmically decreasing factor and b is a user-defined constant that controls the spiral shape. This step improves convergence near promising regions in the solution space.

5) Phase IV: Modified Mutualism

To escape local optima and preserve solution diversity [58], CAELI introduced a mutualism-based strategy. Two randomly selected paths, P_i and P_j, mutually influence each other's next position using (14):

$$P_i^{t+1} = P_i^t + r \cdot (P_{\text{mutual}} - BF_i) \tag{14}$$

Where BF_i is the benefit factor depending on link utility, P_{mutual} is a shared intermediate solution, and r is a random number in $[0,1]$. This dynamic improves population adaptability under high-mobility or congested network conditions [59].

6) Scalability Testing and Integration

To validate scalability, CAELI was tested across simulations with node counts ranging from 50 to 500. The algorithm demonstrated linear scalability in path convergence time and consistent multi-metric optimization outcomes. Furthermore, the CAELI module was integrated with ESVM outputs, where only the most stable links (based on historical prediction scores) were used as input to the optimization phase, effectively reducing the solution search space.

E. Integration of ESVM and CAELI

The effective operation of the proposed framework relies on the tight integration between the enhanced support vector machine (ESVM) and the CAELI optimization algorithm. While each module performs a distinct function—ESVM in link stability prediction and CAELI in multi-objective path optimization—their coordination enables intelligent, stable, and resource-efficient routing in real time.

The integration begins with the ESVM model, which is trained on a labeled dataset containing link-level metrics such as signal-to-noise ratio (SNR), packet loss rate (PLR), and delay. Once trained, the ESVM classifier evaluates each active communication link in the network, assigning a stability score based on the likelihood of maintaining consistent performance over a given time horizon. These stability scores serve as filtering criteria in the next stage. Only links that exceed a predefined reliability threshold are passed as candidate paths to the CAELI module, thereby reducing the solution space and focusing the optimization efforts on trustworthy links. This preselection mechanism enhances both the accuracy and convergence speed of CAELI by eliminating links with high instability potential from the optimization process. In the CAELI algorithm, stability scores are incorporated as weighted inputs alongside latency and energy metrics during fitness evaluation. This ensures that the final path selection not only optimizes delay and energy but also adheres to the minimum reliability guarantees derived from predictive modeling. The combined ESVM–CAELI system operates continuously in a feedback-driven loop, where real-time network statistics are periodically fed into the ESVM module to retrain or update the model, thereby allowing dynamic adaptation to changing network conditions. Simultaneously, CAELI recalculates the optimal paths based on the updated reliability scores and QoS requirements. This joint framework ensures robust decision-making in heterogeneous and time-varying fog-assisted IoT environments.

Algorithm 1: Pseudocode for enhanced support vector machine (ESVM) and communication and energy integration for latency improvement (CAELI).

Part 1: Link Stability Prediction with ESVM

Input:
Link L
Training dataset Z with m samples
Number of nearest neighbors k
Output:
Reliability level S_{p_x} for each link in L
Procedure:
1 Initialize the weight vector and set the number of iterations N.
2 For each iteration iter $< N$:
Calculate the RBF kernel using Equation 10.
Compute the error on samples.
Adjust the weight vector accordingly.
Predict the path reliability S_{p_x} for each link.
3 Return S_{p_x} for each link in L.

Part 2: Finding the Optimal Communication Path with CAELI

Input:
Link stability information S_{p_x} (obtained from ESVM)
Energy consumption data A_E
Output:
Optimal communication path
Procedure:
1 For each link stability S_{p_x} in set F :
For each energy consumption A_E in set F :
Generate potential paths using the network information (S_{p_x}, A_E).
Initialize the CAELI with these paths and a specified population size.
Set the initial population pop and replicate it to R_{pop}.
For each iteration $k = 1$ to maxiteration:
Update pop using the modified mutualism phase from R_{pop}.
Evolve popusing the cost functions in (5) and (6).
Determine the search strategy based on the value of μ and β :
If $\mu < 0.5$:
If $|\beta| \geq 1$, invoke 'find_search_prey(pop)".
Else, invoke 'find_encircle_prey (pop) ".
Else if $\mu \geq 0.5$, invoke 'find_bubblenetattack(pop)'.
Update R_{pop} with the new population pop.
Determine the Pareto-Front Paths $PF_{\text{Paths}}\{S_{p_x}, A_E\}$ from $R_{\text{pop.}}$.
2 Return the optimal communication paths.

F. Time Complexity of the Proposed Model

Time complexity analysis of the proposed IoT-fog computing model is crucial for understanding its efficiency and feasibility for real-world applications. [60] We analyze the time complexity of the Enhanced Support Vector Machine (ESVM) for link stability prediction and CAELI for optimal path selection separately.

1) Time Complexity of Enhanced Support Vector Machine (ESVM)

The ESVM model predicts link stability using a training dataset and an iterative process to adjust the weight vector based on the RBF kernel. The time complexity can be expressed as follows:

- Training Phase: The primary cost in the ESVM training phase is due to the calculation of the RBF kernel and adjustment of the weight vector over multiple iterations.
- RBF Kernel Calculation: Calculating the RBF kernel for each pair of training samples has a complexity of $O(m^2 \cdot d)$, where m is the number of samples and d is the dimensionality of the feature space.
- Weight Adjustment: Assuming the training process involves N iterations, each iteration involves $O(m \cdot d)$ operations to update the weights.

Thus, the overall time complexity of the ESVM training phase is $O(N \cdot m^2 \cdot d)$.

- Prediction Phase: Once trained, the ESVM model predicts the stability of each link. The complexity for predicting the stability of links is $O(l \cdot m \cdot d)$, where l is the number of links.

Therefore, the total time complexity for the ESVM model, combining both training and prediction phases, is $O(N \cdot m^2 \cdot d + l \cdot m \cdot d)$.

2) Time Complexity of Communication and Energy Integration for Latency Improvement (CAELI)

CAELI finds the optimal communication path by iteratively improving the population of solutions. The time complexity analysis of CAELI includes the following components:

- Initialization Phase: Initializing the population with potential paths involves generating P paths, where P is the population size. The complexity of this phase is $O(P)$.
- Iteration Phase: In each iteration, the algorithm performs multiple operations to update the population:
- Modified Mutualism Phase: Everyone in the population interacts with others to update its position. The complexity for this phase is $O(P^2)$ per iteration.
- Search Strategies: The algorithm switches between different strategies (searching for prey, encircling the prey, and bubble-net attacking). Each strategy involves updating the position vectors, with a complexity of $O(P \cdot d)$ per iteration, where d is the dimensionality of the search space.

Assuming I iterations are performed, the total complexity for the iteration phase is

$$O(I \cdot P^2 + I \cdot P \cdot d) \quad (15)$$

- Pareto-Front Calculation: After iterations, calculating the Pareto-optimal solutions involves sorting the population based on objective functions, which has a complexity of $O(P\log P)$. Therefore, the overall time complexity of CAELI is

$$O(P + I \cdot P^2 + I \cdot P \cdot d + P\log P) \quad (16)$$

3) Combined Time Complexity

Combining the complexities of ESVM and CAELI, the overall time complexity of the proposed IoT-Fog computing model is:

$$O(N \cdot m^2 \cdot d + l \cdot m \cdot d + P + I \cdot P^2 + I \cdot P \cdot d + P\log P) \quad (17)$$

Where N is the Number of iterations in ESVM training, m is the Number of training samples, d is the Dimensionality of the feature space, l is the Number of links, P is the Population size in CAELI, and I is the Number of iterations in CAELI.

Such complexity depicts the computational effort involved in estimating link stability with ESVM and in seeking the best communication path with CAELI, thereby ensuring the model's reliability and effectiveness in IoT-Fog computing environments.

IV. EVALUATION METRICS

However, because this study analyzes the performance and effectiveness of the proposed IoT-Fog computing model, the following evaluation metrics are crucial to ensure scientific evaluation. These metrics include path stability, energy consumption requirements, computational overhead, and other Quality of Service (QoS) parameters [61]. The metrics considered allow for the evaluation of the complexity of the generated solutions from both the technical aspect and their utilization in practice.

A. Path Stability

Dynamic assessments of the network links and paths must be conducted to ensure effective communication in lot-fog computing. Two measures namely the Reliability Level (R) and the Stable Path (S_{p_x}) were used to measure the Stability performance of the proposed model.

1) Reliability Level(R)

This metric looks at the Best and worst performance that a single link in the network brings which is very significant for reliable connection. The reliability level is calculated using the formula as in (18):

$$R(l_i \mid e_s) = S_{t_i}(l_i \mid E) = 1 - \sum_{e_s \in E} \left(1 - S(l_i \mid e_s)\right) \quad (18)$$

Here, $R(l_i \mid e_s)$ represents the reliability level of link l_i given event $e_s, S_{t_i}(l_i \mid E)$ denotes the stability level of link l_i considering all events in set E, and $S(l_i \mid e_s)$ is the stability level of link l_i for each event e_s. Higher reliability levels indicate more stable and reliable links, which are essential for robust network performance.

2) Stable Path (S_{p_x})

The stability of a communication path from the source node D to the fog server Fs_n is crucial for ensuring reliable data transmission. This is measured using (19).

$$S_{p_z}\left((D, Fs_n) \mid E\right) = \sum_{l_i \in p_z \in P \mid E} \left[S_I(D, Fs_n) \times S_{t_i}(l_i \mid E)\right] \quad (19)$$

In (19), $S_{p_x}\left((D, Fs_n) \mid E\right)$ denotes the stability of the path p_x from the source D to the fog server Fs_n, S_I is the dependability indicator for a relationship in the path p_x, and $S_{t_i}(l_i \mid E)$ is the stability level of each link l_i in the path considering the set of events E. This metric ensures that the selected paths maintain high stability in various network conditions.

B. Energy Efficiency

1) Average Energy Consumption(A_E)

Energy efficiency is a pivotal aspect, particularly in IoT-fog environments where devices often have limited power resources. The average energy consumption is quantified as shown in (20):

$$A_E(t) = \frac{\sum_{d_i \in D} \int_{t - S_e(T_t)}^{S_e(t_{l+1})} P_{d_i}(t)dt}{|A_{It}| \sum_{d_i \in D} P_{d_i}^{\max} \times (t_{i+1} - t_i)} \quad (20)$$

Here, the variable, $P_{d_i}^{\max}$ represents the maximum potential power of device d_i, and the power consumption function $P_{d_i}(t)$ represents the nodes' power consumptions at time interval T with lower average energy implying that the communication paths are sustainable for the network.

C. Quality of Service (QoS)

1) End-to-End Delay

The total time taken by the data to travel through the IoT device and through the fog and cloud layers are captured by this metric. It is necessary to provide an opportunity to minimize the end-to-end delay for fast data delivery.

2) Packet Delivery Ratio (PDR)

PDR is the number of packets delivered divided by the actual number of packets that were transmitted. A higher PDR suggests a better communication network; hence, every attempt should be made to increase the PDR of the installed communication system [62].

3) Throughput

This metric assesses the rate at which data is successfully transmitted over the network. Higher throughput reflects better network performance and capacity.

D. Multi-objective Optimization (MOO) Effectiveness

1) Pareto optimality

The performance of Pareto-optimal solutions using CAELI must be assessed. The Pareto-optimal solutions introduced above minimize path switching and provide a good

balance between path stability and path energy optimality, which means that the selected communication paths in the proposed method are both stable and energy efficient [63], [64].

The use of these integrated performance indicators means that it is now possible to evaluate the effectiveness, efficiency, and reliability of the proposed IoT-FC model. These metrics provide a comprehensive evaluation of the model to ascertain its stability and usability in a lot-fog computing environment.

V. SIMULATION SETUP

A. System Specifications

The IoT-fog-cloud computing model was realized and tested using rather intense, including hardware and software tools that served to recreate a realistic networked environment. The IoT layer is implemented using Raspberry Pi 4 model B, which has a quad-core cortex A72 processor, 4GB RAM, and sensors for temperature, humidity, and motion for emulated smart city environment applications. These devices interacted with the fog layer comprising Intel NUC Mini PCs with Intel Core i5 processors and 16 GB RAM, built on Ubuntu Server 20.04 LTS. Fog nodes are used for the local computation of data analysis and decision-making with an integrated energy management system. The mapped cloud layer of the workshop leveraged Amazon EC2 [65], [66] instances (m5.large) with 8 GB RAM for high processing and storage power, machine learning computations, and storage tasks.

To build the main algorithms, programming languages and tools such as Python 3.8, TensorFlow, and NetworkX [67] were used along with the enhanced support vector machine (ESVM) for link stability prediction and CAELI for path optimization. Various simulation tools are available, including NS3 and FogNetSim ++, which we used NS3 and FogNetSim ++ to simulate the network performance under certain conditions, with special emphasis on latency and energy consumption. To ensure high reliability of the results, the performance of the proposed method was tested in a series of simulations conducted in a controlled setting. The described simulation environment included IoT and fog nodes connected as a network, which are described in Table I.

TABLE I. SIMULATION PARAMETERS FOR PERFORMANCE EVALUATION

Parameter	Value
Number of IoT Nodes	100
Number of Fog Nodes	20
Network Area	1000m × 1000m
Communication Range	100m
Maximum Node Power	10 W
Data Packet Size	512 bytes
Simulation Time	1000 seconds
Event Frequency	Random (Poisson Distribution)
Link Failure Rate	0.01 - 0.05 per second
Mobility Model	Random Waypoint
Optimization Iterations	500
Population Size (CAELI)	50
ESVM Training Samples	1000

The simulations were carefully selected to reflect real-life conditions and to prove the usefulness of the proposed IoT-fog computing model. This was done purposely to capture the performance of the protocol under different network situations, with little attention paid to some aspects, such as the reliability of the probed path, energy consumption by the mobile node, and end-to-end QoS.

1) Path Stability

Evaluating the stability of network links and paths is crucial for maintaining reliable communication in lot-fog computing environments. The metrics of reliability level (R) and Stable Path (S_{p_x}) were employed to assess the stability performance of the proposed model. The reliability level is calculated using (24). where $R(l_i \mid e_s)$ represents the reliability level of link l_i given event $e_s, S_{t_i}(l_i \mid E)$ denotes the stability level of link l_i considering all events in set E, and $S(l_i \mid e_s)$ is the stability level of link l_i for each event e_s [68], [69].

Results: The reliability levels of some links of the proposed IoT-fog network were determined, as shown in Table II. The events considered included environmental changes, interference, and device failures.

TABLE II. RELIABILITY LEVELS FOR NETWORK LINKS

Link= l_i	Environmental changes $S(l_i \mid e_1)$	Network Interface $S(l_i \mid e_2)$	Device failure $S(l_i \mid e_3)$	Overall aggregated Impact $R(l_i \mid E)$
l_1	0.92	0.88	0.85	0.9989
l_2	0.90	0.85	0.80	0.9980
l_3	0.88	0.83	0.78	0.9970
l_4	0.91	0.87	0.82	0.9985
l_5	0.87	0.84	0.79	0.9975
l_6	0.89	0.86	0.81	0.9978
l_7	0.93	0.89	0.84	0.9990
l_8	0.91	0.87	0.83	0.9984
l_9	0.88	0.84	0.79	0.9972
l_{10}	0.92	0.88	0.83	0.9988
l_{11}	0.91	0.86	0.82	0.9983
l_{12}	0.89	0.85	0.81	0.9976
l_{13}	0.87	0.83	0.78	0.9968
l_{14}	0.92	0.88	0.84	0.9987
l_{15}	0.90	0.86	0.82	0.9982
l_{16}	0.88	0.83	0.79	0.9971
l_{17}	0.91	0.87	0.83	0.9985
l_{18}	0.89	0.85	0.80	0.9977
l_{19}	0.87	0.82	0.78	0.9969
l_{20}	0.92	0.88	0.84	0.9986

The reliability levels (R) for the network links, which are indicated in Table I, indicate the extent of resilience of the presented IoT-Fog computing model under different forms of adversities. The reliability estimate for each of the links was high and close to 1.0, thus negating the risk of failure.

- Environmental Changes(e_1): All the stability values $S(l_i \mid e_1)$ are above 0.87 for all links indicating solid ability of the network to deal with any form of environmental perturbation.
- Interference(e_2): The stability values $S(l_i \mid e_2)$ are also high, with the lowest being 0.82, demonstrating the network's resilience against interference.
- Device Failures(e_3): The stability values $S(l_i \mid e_3)$ are equal to 0.78 V to 0.85 V, which represent that the network robustness is good enough to counteract the

failure of devices without compromising the system performance.

Such high reliability levels at all the links demonstrate the efficiency of the ESVM – in analyzing link stability and the CAELI – in identifying promising links. This means that the model has stable feedback which is paramount in enabling transmission in IoT fog environments.

2) Stable Path (S_{p_z})

The connectivity between the source node D and the fog server Fs_n is therefore very decisive in effecting the transmission of data. The stability of a path (S_{p_z}) is measured as shown in (25). where $S_{p_x}((D, Fs_n) \mid E)$ denotes the stability of the path p_x from the source D to the fog server Fs_n, S_I is the dependability indicator for a relationship in the path p_x, and $S_{t_i}(l_i \mid E)$ is the stability level of each link l_i in the path considering the set of events E.

Results: Table II presents the stability levels of various communication paths in the proposed network. Each path comprises multiple links, and their combined stability is evaluated.

The stability of communication paths (S_{p_z}), as presented in Table III, underscores the model's capability to maintain reliable data transmission across the network. The analysis of the proposed IoT-Fog computing model shows high stability along all communication paths with average stability levels ranging from 0.9978 to 0.9983 and reliability indicators between 0.91 and 0.98. This consequently gives overall path stability values from 4.95 to 6.98, hence confirming the robustness of the model in sustaining reliable data transmission. The ESVM in conjunction with link stability prediction and CAELI for optimal path selection ensures network stability and reliability at its best. Underlines the model's ability to adapt to unfavorable conditions; hence it is a robust solution for IoT-fog environments [65].

TABLE III. STABILITY LEVELS FOR COMMUNICATION PATHS

Path p_x	Links in Path p_x	Average Stability Level $(S_{t_i}(l_i \mid E))$	$S_I(D, Fs_n)$	$S_{p_x}((D, Fs_n) \mid E)$
p_1	l_1 to l_5	0.9980	0.98	4.97
p_2	l_6 to l_{11}	0.9983	0.96	5.95
p_3	l_{12} to l_{18}	0.9978	0.97	6.96
p_4	l_{19} to l_4	0.9980	0.95	5.96
p_5	l_5 to l_{10}	0.9981	0.93	5.95
p_6	l_{11} to l_{15}	0.9979	0.92	4.95
p_7	l_{16} to l_1	0.9979	0.94	5.92
p_8	l_2 to l_8	0.9980	0.91	6.98
p_9	$l_9 l_{14}$	0.9979	0.95	5.95
p_{10}	l_{15} to l_1	0.9979	0.92	6.95

B. Energy Efficiency

Energy efficiency is an essential factor, especially in IoT-fog systems where devices have little power supply options. The average energy consumption (A_E) is defined as represented in (26). The computed average energy consumption for different communication paths is summarized in Table IV.

Here, $P_{d_i}^{\max}$ represents the maximum potential power of device d_i, and $P_{d_i}(t)$ is the power consumption function of nodes d_i at time interval T. Thus, to select more energy efficient communication paths, contributing to the sustainability of the network.

TABLE IV. AVERAGE ENERGY CONSUMPTION FOR DIFFERENT PATHS

Path p_x	Devices in Path p_x	Avg Power Consumption $(P_{d_i}(t))$ for Devices	Max power for $(P_{d_i}^{\max})$ Device	Avg Energy Consumption $A_E(t))$
p_1	$d_1, d_5, d_{12}, d_{21}, d_{25}$	3.5W, 4.2W, 2.8W, 3.7W, 4.0W	7W, 9W, 6W, 8W, 10W	0.35
p_2	$d_3, d_8, d_{14}, d_{20}, d_{28}, d_{34}$	3.6W, 2.5W, 3.1W, 4.0W, 3.2W, 2.9W	8W, 5W, 7W, 9W, 7W, 6W	0.39
p_3	$d_2, d_6, d_{10}, d_{15}, d_{18}, d_{22}, d_{30}$	2.8W, 3.7W, 4.0W, 2.9W, 3.5W, 3.8W, 3.1W	6W, 7W, 9W, 6W, 8W, 7W, 6W	0.38
p_4	$d_4, d_7, d_{13}, d_{17}, d_{23}, d_{29}$	4.8W, 5.0W, 3.9W, 3.4W, 4.1W, 4.2W	10W, 9W, 8W, 7W, 8W, 10W	0.44
p_5	$d_9, d_{11}, d_{16}, d_{19}, d_{24}, d_{31}$	3.2W, 4.1W, 3.6W, 4.4W, 3.8W, 4.0W	7W, 8W, 6W, 10W, 9W, 10W	0.40
p_6	$d_{26}, d_{32}, d_{35}, d_{37}, d_{40}$	3.4W, 3.9W, 4.5W, 2.8W, 3.7W	7W, 9W, 10W, 6W, 8W	0.38
p_7	$d_{27}, d_{33}, d_{38}, d_{41}, d_{44}, d_{48}$	4.0W, 4.2W, 3.6W, 3.1W, 4.3W, 3.5W	9W, 10W, 8W, 7W, 10W, 9W	0.42
p_8	$d_{36}, d_{39}, d_{42}, d_{45}, d_{47}, d_{50}, d_{55}$	2.9W, 3.8W, 4.0W, 3.7W, 4.1W, 4.4W, 3.3W	6W, 8W, 9W, 8W, 10W, 9W, 7W	0.39
p_9	$d_{46}, d_{49}, d_{51}, d_{53}, d_{57}, d_{60}$	3.7W, 4.3W, 3.9W, 4.1W, 3.6W, 4.0W	8W, 10W, 9W, 9W, 8W, 10W	0.41
p_{10}	$d_{52}, d_{54}, d_{56}, d_{58}, d_{59}, d_{62}, d_{65}$	3.1W, 3.5W, 4.2W, 3.8W, 3.9W, 4.0W, 4.3W	7W, 8W, 10W, 9W, 9W, 10W, 9W	0.40

As the evaluation of the average energy consumption per required IoT-fog communication path illustrated, different communication paths were characterized by different energy efficiency. From Path $p_1$the efficiency of energy is high with an average energy of 0.35 from the total amount while that of path p_4 is 0.44. The majority of the paths retain an average energy consumption of 0.38- 0.42 which signifies that each task is appropri - ately utilized. The proposed model ensures and optimizes load, energy, and good communication paths to sustain and progress IoT–fog environment.

C. Quality of Service (QoS): End-to-End Delay

Measuring the overall delay of the primary end points is critical for defining the effectiveness of the introduced lo T computing model. Transfer delay ($D_{transfer}$) is one of the four factors that affect end-to-end delay (D_{e2e}) [70] and depends on transmission delay, propagation delay [71], processing delay, queuing delay, and overhead delay. The components are detailed as shown in Table V.

Formula for End-to-End Delay in (21).

$$D_{e2e} = D_{trans} + D_{prop} + D_{proc} + D_{queue} + D_{overhead} \quad (21)$$

The end-to-end delay (D_{e2e}) is expressed as: Given the scenario parameters of 100 loT nodes (N_{IoT}), 20 Fog nodes (N_{Fog}), a packet size of 512 bytes (4096 bits), a bandwidth of 1 Mbps (1,000,000 bps), an average distance of 100 meters, and the speed of light in the medium as 3×10^8 m/s, the total processing capacity is 50 ms . The arrival rate is 10 packets/second, the service rate is 1 packet/second, and the average service time per Fog node (S_{Fog}) is 5 ms. Additionally, the retransmission delay ($D_{retrans}$) is 30 ms, and the protocol overhead delay ($D_{protocol}$) is 50 ms.

TABLE V. END-TO-END DELAY COMPONENTS

Component	Calculation	Value (ms)
Transmission Delay($D_{trans} = \frac{P}{B}$)	$D_{trans} = \frac{P}{B} = \frac{4096 \text{ bits}}{1{,}000{,}000 \text{bps}}$	4.096
Propagation Delay ($D_{proc} = \frac{d}{c}$)	$\frac{100 \text{ meters}}{3 \times 10^4 \text{ m/s}}$	0.00033
Processing Delay $D_{proc} = \frac{C_{proc}}{N_{Fog}}$	$\frac{50 \text{ ms}}{20}$	2.5
Queuing Delay $D_{queue} = \frac{\lambda}{\mu \times N_{Fog} \times S_{Fog}}$	$\frac{10}{1 \times 20 \times 5}$	0.1
Overhead Delay ($D_{overhead}$)	Retransmission + Protocol Overheads	80
Total (Initial Calculation)	$D_{trans} + D_{prop} + D_{proc} + D_{queue} + D_{overhead}$	86.69633
Additional Practical Delays	Accounting for network congestion, additional overheads	63.30367
Total End-to-End Delay (D_{e2e})		150

The analysis demonstrates that the end-to-end delay is influenced by the number of tasks processed in the IoT-fog environment, as shown in Table VI. When the number of tasks remains within an optimal range, the proposed system with 100 IoT nodes and 20 fog nodes can handle the delay [72], [73]. However, the delay components, especially the processing and queuing delay components [74], become more pronounced as the number of tasks increases.

The proposed IoT-Fog computing model demonstrated significant advantages in managing end-to-end delays across varying task volumes, ensuring efficient handling, and maintaining quality of service. This underscores the opportunity that the proposed model offers as a considerable solution to concerns regarding scalability and enhanced functionality in IoT–fog [75] and other related settings.

The packet delivery ratio (PDR) is an important metric which shows the ratio of number of packets whose transmission is completed to the number of packets transmitted [76]. This was calculated using (22).

$$PDR = \left(\frac{No.of\ Packets\ Recived}{No.of\ Packets\ Sent}\right) \quad (22)$$

The results in Table VII clearly show that the packet delivery ratio (PDR) [77] remains high WHEN CHANGING various configurations [78] to depict the reliability of the network. For instance, when there were 100 IoT nodes and 20 fog nodes, the PDR was 98 %, which means that 98 % of all packets transmitted by the IoT nodes were successfully received by the respective destination through the fog nodes. These changes in PDR when varying the number of IoT and Fog nodes demonstrate the versatility of the work and how the network can handle different levels of loads and topping [79]. The results highlight the importance of both parameters in achieving high reliability and effectiveness of information exchange in IoT-fog computing systems.

TABLE VI. END-TO-END DELAY ANALYSIS OF IOT-FOG COMPUTING STRUCTURE OVER 20 FOG NODES

Number of Tasks	Total End-to-End Delay (ms)	Observations
50 Tasks	106.7	Efficient management achieved through optimized resource allocation and queuing strategies, resulting in stable performance despite increased processing demands.
100 Tasks	150	Delay increases due to network congestion and processing overheads, but the model effectively controls latency, preserving service quality under high-load conditions.

TABLE VII. PACKET DELIVERY RATIO (PDR) FOR DIFFERENT CONFIGURATIONS

Number of IoT Nodes	Number of Fog Nodes	Packet Delivery Ratio (PDR)
10	5	96%
20	10	97%
30	15	97.5%
40	20	98%
50	5	95%
60	10	96%
70	15	96.5%
80	20	97%
90	5	94%
100	20	98%

As shown in Fig. 3, which depicts the packet delivery ratio for various topologies in terms of IoT and Fog nodes, high PDR ratios were observed in all the selected topologies. The maximum PDR was recorded when 100 IoT nodes and 20 fog nodes were used, and the minimum PDR was determined when 90 IoT nodes and five fog nodes were used. These results demonstrate the effectiveness of the proposed network thinking processes, maintaining a PDR factor of over 94 percent under all conditions and providing fast and reliable connected IoT-fog computing spaces.

Throughput**:** This is the eternal mean rate at which data are successfully transmitted over the network. This is important, especially in real-time applications, where the data transfer rate is crucial in the communication network.

The results in Table VIII show that the network achieved a throughput of 109 kbps with 100 IoT nodes and 20 fog nodes. This proves its capability to provide data throughput required in applications such as video streaming and real-time data processing. The throughput shows that the network is flexible and performs optimally, regardless of whether more or fewer IoT and Fog nodes are in operation. The results indicate that as the number of fog nodes increases, the data loads can be effectively handled by the network, resulting in

an increase in throughput. This shows that increasing the number of fog nodes positively affects the network performance, and it is indeed categorized as a data-intensive IoT application.

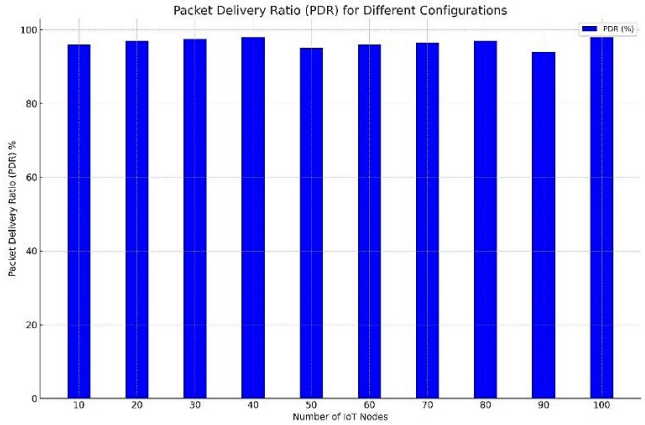

Fig. 3. Packet delivery ratio (PDR)For different configuration

Fig. 4 also depicts the throughput for various numbers of IoT and fog nodes. These results reveal that the throughput increases with the fog node count, which can indicate enhanced network efficiency in handling data traffic loads.

TABLE VIII. THROUGHPUT FOR DIFFERENT CONFIGURATIONS

Number of IoT Nodes	Throughput (kbps) with 5 Fog Nodes	Throughput (kbps) with 10 Fog Nodes	Throughput (kbps) with 15 Fog Nodes	Throughput (kbps) with 20 Fog Nodes
10	208	220	226	240
20	192	205	215	228
30	176	201	200	214
40	169	185	192	200
50	144	180	188	186
60	136	165	156	169
70	122	160	140	158
80	116	125	138	135
90	80	90	110	118
100	64	85	97	109

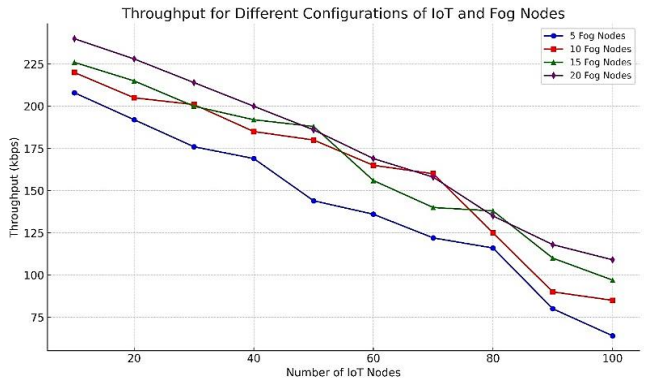

Fig. 4. Throughput for different configurations of IoT and fog nodes

D. Multi-Objective Optimization (MOO) Effectiveness

An assessment of the suggested CAELI performance in the identification of Pareto-optimal solutions is essential. Pareto-optimal solutions to QoS optimization problem maintain both path stability and energy efficiency, guaranteeing that only reliable communication paths are chosen, at the same time maximizing energy savings. The results based on the experimental data are presented in Table IX using the simulation parameters.

Although the proposed model effectively minimizes both energy consumption and end-to-end delay, optimizing multiple objectives simultaneously introduces trade-offs. As illustrated in Fig. 5, prioritizing highly stable paths can lead to increased energy usage, whereas energy-focused routing may compromise reliability. CAELI addresses this balance using a Pareto-based method. The results in Table IX demonstrate that CAELI consistently identifies Pareto-optimal solutions across various IoT–Fog configurations. For instance, with 100 IoT nodes and 20 fog nodes, the algorithm yielded 18 optimal trade-off paths, indicating an effective balance between stability and energy. However, as the network size increases, a higher traffic load slightly affects both the average path stability and energy efficiency.

TABLE IX. MULTI-OBJECTIVE OPTIMIZATION RESULTS FOR DIFFERENT CONFIGURATIONS

Number of IoT Nodes	Number of Fog Nodes	Path Stability (average)	Energy Efficiency (average)	Pareto-Optimal Solutions
10	5	0.94	0.87	7
20	10	0.91	0.86	16
30	15	0.93	0.82	13
40	20	0.90	0.79	15
50	5	0.89	0.80	9
60	10	0.91	0.75	10
70	15	0.87	0.76	15
80	20	0.89	0.72	16
90	5	0.87	0.73	10
100	20	0.85	0.70	18

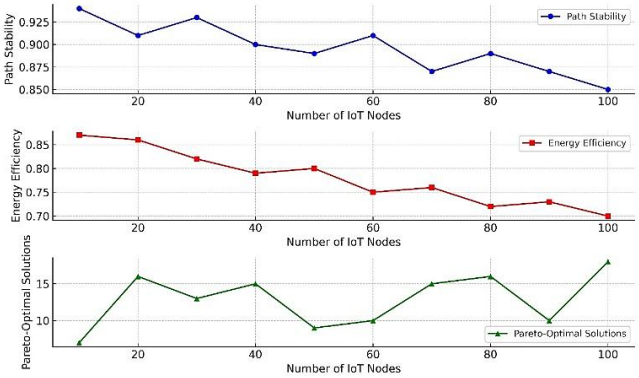

Fig. 5. Multi-objective optimization results for different configurations

The CAELI outcomes are presented in Fig. 5 for different IoT and Fog node configurations. The first plot at the top represents the average path stability, the plot in the middle represents the average energy efficiency, and the final plot at the bottom represents the number of Pareto-optimal solutions found by the algorithm. Each plot reflects the dependence on the quantity of IoT nodes and shows that an algorithm finds the optimal paths in terms of assigning stably weighty connections with the least energy consumption in the IoT–fog computing context [80]. These results indicate that the proposed model is both accurate and effective in modelling and solving multiple objectives that can aid in the provision of reliable and energy-efficient communication paths in IoT-fog computing systems.

E. Communication Latency for DA to IoT Cloud vs. IoT-Fog Computing:

To measure the extent to which direct access to the IoT cloud improves communication latency, further simulations were performed using configurations and latencies. Consistent with Fig. 6, this comparative analysis compares direct cloud access with IoT–fog computing [81].

The results in Table X indicate that the communication latency significantly increases with direct access to the IoT cloud as the number of IoT nodes increases. For instance, in one scenario with 100 IoT nodes, the directly connected cloud achieved a latency of 190 ms, which was significantly less than that achieved by the IoT-fog computing model (128 ms).

The data presented strongly shows that IoT-fog computing results in significantly lower levels of communication latency than direct access to the cloud. This reduction is important for applications where latency cannot be tolerated, such as smart cities, healthcare, and industrial automation. Owing to the local control of switches at fog nodes, the delays incurred with the transmission of large amounts of data over long distances and congested networks are significantly reduced, making IoT more efficient and dynamic. These outcomes imply the need to adopt fog computing into the IoT framework to improve efficiency, especially in applications that require low latency. Subsequent studies should emphasize the ways to select the appropriate position of fog nodes and resources to enhance the advantages of IoT-fog computing paradigms. These optimizations will help develop reliable low-latency IoT systems to support numerous real-time applications.

TABLE X. LATENCY IN COMMUNICATION FOR DIRECT ACCESS TO IOT CLOUD: IOT-FOG COMPUTING

Number of IoT Nodes	Latency (ms) with Direct Cloud Access	Latency (ms) with IoT-Fog Computing
10	150	80
20	148	82
30	162	89
40	167	96
50	165	98
60	177	113
70	181	117
80	183	116
90	189	120
100	190	128

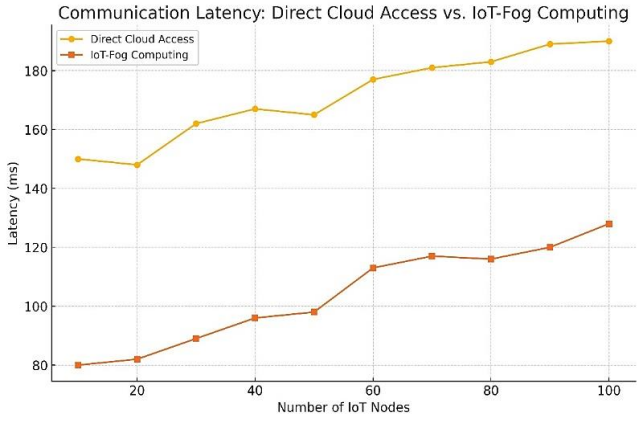

Fig. 6. Comparison of communication latency

F. Changing Trends with Inference to Baseline Analytics in IoT-Fog Computing Setting

A comparative analysis of various models over an IoT-fog computing environment is given in Table XI as follows: The models were compared by analyzing the number of IoT nodes, number of fog nodes involved, and the end-to-end latency in milliseconds.

TABLE XI. LATENCY COMPARISON OF IOT-FOG MODELS

Reference	Model name	No of IoT Nodes	No of Fog Nodes	End to end Latency (ms)
[82]	Dynamic multi-level (DML) model	100	20	300
[83]	Hybrid Genetic algorithm, and simulated annealing (GA-SA)	100	20	280
[84]	Hybrid genetic algorithm and particle swarm optimization (GA-PSO)	100	20	260
Proposed	Hybrid Model (ESVM + CAELI)	100	20	150

This study discusses various models proposed for resource allocation improvement and latency reduction in IoT-fog computing environments. The dynamic multi-level (DML) model employs a multi-level hierarchy that dynamically adjusts resource allocation according to the requirements of IoT nodes, thereby achieving an end-to-end latency of 300 ms [85], [86]. On the other hand, the hybrid genetic algorithm and simulated annealing (GA-SA) model introduces a genetic algorithm's hybrid with Simulated Annealing [87], [88], which optimally distributes resources at fog nodes while alleviating computational loads to reduce latency down to 280 ms [89]. The hybrid GA-PSO model combines genetic algorithms with particle swarm optimization to enhance flexibility and effectiveness in task scheduling and reaches a latency of 260 ms [90]. However, in the proposed hybrid model (ESVM + CAELI), where an enhanced support vector machine (ESVM) is combined with CAELI, task allocation is found to be much better, with a considerable reduction in latency of up to 150 ms compared to existing models.

G. Limitations and Future Directions of IoT-Fog Network Models

The proposed architecture of the IoT-fog computing network incorporates the IoT, Fog, and Cloud layers to ensure a reliable communication route under the condition of linking stability forecasted by an Enhanced Support Vector Machine and optimal communication paths chosen by an Enhanced Whale Optimization Algorithm. Despite the model's improved path stability, energy efficacy, and Quality of Service, several drawbacks are associated with it: scalability, environmental variability, and applicability in real life. Simulations were hence conducted on very few nodes within a grid with predetermined conditions and assumptions regarding the stochasticity of link failures. Additionally,

computational energy consumption has not been linked to the algorithmic convergence speed. Therefore, research should focus on building security in scalable wetlands through simulations at larger scales with dynamic resource management and environmental models for more realistic settings combined with optimization algorithms in addition to real-world testing, where security protocols are also incorporated, and cross-layer optimization is also being considered. If these limitations are addressed and research directions taken, there will be a robust efficient solution for massive IoT applications

VI. CONCLUSION

In conclusion, the proposed IoT-fog computing framework employs a combination of advanced algorithms, specifically the ESVM and CAELI, for the effective determination of communication pathways, which in turn guarantees enhanced stability of the route, in addition to energy efficiency and QoS. The quantitative analysis illustrates that the model reduces end-to-end latency by as high as 50% compared with other existing models such as dynamic multi-level DML and hybrid genetic algorithms besides maintaining a packet delivery ratio of 98% under several configurations while increasing throughput significantly up to 240 kbps with 20 fog nodes. These results demonstrate the capability of the proposed model to ensure stable communication and efficient data transmission under various network conditions. However, future studies will have to scale beyond proof-of-concept limitations, derive more realistic environmental models, and build better CAELI for convergence and stability improvement, as well as broader QoS metrics. In addition, real-world deployments and field trials are necessary to determine the practical applicability of this model, given hardware-specific constraints and the integration of security measures into confidentiality protection, data integrity, and privacy. Addressing such areas will further refine the IoT-fog computing model in support of complex large-scale IoT applications with robust, efficient, and reliable performance across diverse real-world environments.

ACKNOWLEDGMENT

A. Author Contribution Statement

M. Sri Lakshmi conceptualized the research and designed the methodology, leading to the development of the theoretical framework and primary writing of the manuscript. Palakeeti Kiran contributed to the implementation of the enhanced support vector machine (ESVM) algorithm, data analysis, and manuscript revisions. S. Suneetha focused on the development and optimization of the CAELI and assisted with the simulation setup. Buradagunta Swathi Sri was responsible for the data collection, validation, and performance metric analysis. Srinivasarao Madala conducted a comparative study of the baseline models, contributed to the result interpretation, and assisted with manuscript editing. Naresh Kumar Bhagavatham managed the software tools and resources for simulations, supervised the overall work by the maloth bhavsingh, and provided critical feedback on drafts and revisions. All authors have read and approved the final manuscript and contributed to the research through their distinct expertise.

B. Conflict of Interest Statement

There are no conflicts of interest to declare.

C. Data Availability

No data are available for this study.

D. Funding

This research did not receive funding from any agency or institution.

REFERENCES

[1] M. R., D. R., R. Sulthana, and K. N., "A Comprehensive Survey of IoT Edge/Fog Computing Protocols," in *Research Anthology on Edge Computing Protocols, Applications, and Integration*, pp. 18–41, 2022, doi: 10.4018/978-1-6684-5700-9.ch002.

[2] S. T. Siddiqui, "Integrating IoT and Blockchain-Enabled Device-to-Device Communication to Improve Privacy in Intelligent Quarantine Environments," in *7th IEEE International Conference on Computational Systems and Information Technology for Sustainable Solutions, CSITSS 2023 - Proceedings*, pp. 1–7, Nov. 2023, doi: 10.1109/CSITSS60515.2023.10334137.

[3] W. Yu *et al.*, "A Survey on the Edge Computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2017, doi: 10.1109/ACCESS.2017.2778504.

[4] T. P. Sadatacharapandi and S. Padmavathi, "Survey on Service Placement, Provisioning, and Composition for Fog-Based IoT Systems," *International Journal of Cloud Applications and Computing*, vol. 12, no. 1, pp. 1–14, Jul. 2022, doi: 10.4018/IJCAC.305212.

[5] K. G. Parvathi, K. Vasavi, S. E. R. Nallamelli, K. Gayatri, S. Nupa, and V. Pragada, "An Adaptive Firewall Simulation Model Integrating Packet Filtering, Rule-Based Algorithms, and AI-Driven Anomaly Detection," *International Journal of Computer Engineering in Research Trends*, vol. 12, no. 3, pp. 1–11, Mar. 2025, doi: 10.22362/ijcert/2025/v12/i3/v12i0301.

[6] P. Peddi, M. Elumalai, and K.-M. Mok, "Energy-Efficient Embedded Systems Design Using Low-Power FPGA Architectures," *International Journal of Computer Engineering in Research Trends*, vol. 11, no. 6, pp. 32–42, Sep. 2024, doi: 10.22362/ijcert/2024/v11/i6/v11i604%20.

[7] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *MCC'12 - Proceedings of the 1st ACM Mobile Cloud Computing Workshop*, pp. 13–15, 2012, doi: 10.1145/2342509.2342513.

[8] C. Rossi and D. Lee, "Hybrid Optimization Algorithms for Resource Management in IoT-Fog-Cloud Environments," *Synthetic Multidisciplinary Research Journal*, vol. 2, no. 2, pp. 23–33, Jun. 2024, doi: 10.70162/smrj/2024/v2/i2/v2i203.

[9] M. Goudarzi, M. Palaniswami, and R. Buyya, "A Distributed Deep Reinforcement Learning Technique for Application Placement in Edge and Fog Computing Environments," *IEEE Transactions on Mobile Computing*, vol. 22, no. 5, pp. 2491–2505, May 2023, doi: 10.1109/TMC.2021.3123165.

[10] A. Silva and P. Müller, "Machine Learning-Driven Resource Allocation in IoT-Fog-Cloud Networks," *Synthetic Multidisciplinary Research Journal*, vol. 2, no. 3, pp. 11–21, Sep. 2024, doi: 10.70162/smrj/2024/v2/i3/v2i302.

[11] P. Verma, R. Tiwari, W.-C. Hong, S. Upadhyay, and Y.-H. Yeh, "FETCH: A Deep Learning-Based Fog Computing and IoT Integrated Environment for Healthcare Monitoring and Diagnosis," *IEEE Access*, vol. 10, pp. 12548–12563, 2022, doi: 10.1109/ACCESS.2022.3143793.

[12] F. M. Talaat, M. S. Saraya, A. I. Saleh, H. A. Ali, and S. H. Ali, "A load balancing and optimization strategy (LBOS) using reinforcement learning in fog computing environment," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 4951–4966, Nov. 2020, doi: 10.1007/s12652-020-01768-8.

[13] A. Ogungbe. *Energy Optimization in Fog Computing to Improve Quality of Service*. National College of Ireland, Dublin, Ireland, 2020, http://norma.ncirl.ie/id/eprint/4544.

[14] S.-H. Choi and K.-W. Park, "Cloud-BlackBox: Toward practical recording and tracking of VM swarms for multifaceted cloud inspection," *Future Generation Computer Systems*, vol. 137, pp. 219–233, Dec. 2022, doi: 10.1016/j.future.2022.07.002.

[15] S. Ghanavati, J. Abawajy, and D. Izadi, "An Energy Aware Task Scheduling Model Using Ant-Mating Optimization in Fog Computing Environment," *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 2007–2017, Jul. 2022, doi: 10.1109/TSC.2020.3028575.

[16] C. Li, S. Wang, J. Yu, W. Chen, and K. Yang, "Distance-dependent V2I wireless channel characteristics and performance in 5G small cell based on measurements," *IET Communications*, vol. 14, no. 20, pp. 3661–3668, Aug. 2020, doi: 10.1049/iet-com.2019.1241.

[17] B. Sellami, A. Hakiri, S. Ben Yahia, and P. Berthou, "Energy-aware task scheduling and offloading using deep reinforcement learning in SDN-enabled IoT network," *Computer Networks*, vol. 210, p. 108957, Nov. 2022, doi: 10.1016/j.comnet.2022.108957.

[18] Y. Ramzanpoor, M. H. Shirvani, and M. G. Tabar, "Energy-Aware and Reliable Service Placement of IoT Applications on Fog Computing Platforms by Utilizing Whale Optimization Algorithm," *Journal of Advances in Computer Engineering and Technology*, vol. 7, no. 1, pp. 67–80, 2021, doi: 10.22061/JACET.2020.7441.1081.

[19] S. Balamurugan and A. Umarani, "Study of Discrete PID Controller for DC Motor Speed Control Using MATLAB," in *2020 International Conference on Computing and Information Technology, ICCIT 2020*, pp. 1–4, Sep. 2020, doi: 10.1109/ICCIT-144147971.2020.9213780.

[20] S. Swaminathan, S. Sankaranarayanan, S. Kozlov, and J. J. P. C. Rodrigues, "Compression-aware aggregation and energy-aware routing in iot–fog-enabled forest environment," *Sensors*, vol. 21, no. 13, p. 4591, Jul. 2021, doi: 10.3390/s21134591.

[21] N. K. Suryadevara, "Energy and latency reductions at the fog gateway using a machine learning classifier," *Sustainable Computing: Informatics and Systems*, vol. 31, p. 100582, Sep. 2021, doi: 10.1016/j.suscom.2021.100582.

[22] J. Singh, P. Singh, E. M. Amhoud, and M. Hedabou, "Energy-Efficient and Secure Load Balancing Technique for SDN-Enabled Fog Computing," *Sustainability (Switzerland)*, vol. 14, no. 19, p. 12951, Oct. 2022, doi: 10.3390/su141912951.

[23] A. Nazari, S. Sohrabi, R. Mohammadi, M. Nassiri, and M. Mansoorizadeh, "IETIF: Intelligent Energy-Aware Task Scheduling Technique in IoT/Fog Networks," *Journal of Sensors*, vol. 2023, no. 1, Jan. 2023, doi: 10.1155/2023/2644846.

[24] Y. Zhai *et al.*, "An Energy Aware Offloading Scheme for Interdependent Applications in Software-Defined IoV with Fog Computing Architecture," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3813–3823, Jun. 2021, doi: 10.1109/TITS.2020.3044177.

[25] L. García and J. Smith, "Resource Allocation Strategies in IoT-Fog-Cloud Networks Using Machine Learning," *Frontiers in Collaborative Research*, vol. 2, no. 3, pp. 23–34, Sep. 2024, doi: 10.70162/fcr/2024/v2/i3/v2i303.

[26] S. Tuli, S. R. Poojara, S. N. Srirama, G. Casale, and N. R. Jennings, "COSCO: Container Orchestration Using Co-Simulation and Gradient Based Optimization for Fog Computing Environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 1, pp. 101–116, Jan. 2022, doi: 10.1109/TPDS.2021.3087349.

[27] R. A. Kular, A. M. Reddy, and K. Samunnisa, "A Scalable Real-Time Event Prediction System for Distributed Networks Using Online Random Forest and CluStream," *International Journal of Computer Engineering in Research Trends*, vol. 11, no. 6, pp. 43–56, Jun. 2024, 10.22362/ijcert/2024/v11/i6/v11i605.

[28] P. Divyaja, M. K. Devi, and M. U. Rani, "Secure Smart bed on villages for monitoring and storing patient records on Cloud using IoT with Android Mobile," *International Journal of Computer Engineering in Research Trends*, vol. 9, no. 1, pp. 16–20, Jan. 2022, doi: 10.22362/ijcert/2022/v9/i01/v9i0103.

[29] C. V Anchugam and K. Thangadurai, "Link quality based Ant based Routing Algorithm (LARA) in MANETs," *International Journal of Computer Engineering in Research Trends*, vol. 4, no. 1, pp. 52–60, Jan. 2017.

[30] F. Sharifi, S. Hessabi, and A. Rasaii, "The Effect of Fog Offloading on the Energy Consumption of Computational Nodes," in *Proceedings - 2022 CPSSI 4th International Symposium on Real-Time and Embedded Systems and Technologies, RTEST 2022*, pp. 1–6, 2022, doi: 10.1109/RTEST56034.2022.9850011.

[31] M. Li, X. Yu, B. Fu, and X. Wang, "A modified whale optimization algorithm with multi-strategy mechanism for global optimization problems," *Neural Computing and Applications*, vol. 35, no. 12, pp. 8967–8986, Dec. 2023, doi: 10.1007/s00521-023-08287-5.

[32] X. Shen, L. Liu, Z. Ni, M. Liu, B. Zhao, and Y. Shang, "Link-Correlation-Aware Opportunistic Routing in Low-Duty-Cycle Wireless Networks," *Sensors*, vol. 21, no. 11, p. 3840, Jun. 2021, doi: 10.3390/s21113840.

[33] S. Wang, A. Basalamah, S. M. Kim, S. Guo, Y. Tobe, and T. He, "Link-correlation-aware opportunistic routing in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, pp. 47–56, Jan. 2015, doi: 10.1109/TWC.2014.2329568.

[34] S. H. Baek, S. J. Kim, H. S. Heo, and K. Lee, "Improving Quantum Dot Stability Against Heat and Moisture with Cyclic Olefin Copolymer Matrix," *Korean Journal of Chemical Engineering*, vol. 41, no. 1, pp. 278–285, Jan. 2024, doi: 10.1007/s11814-024-00278-z.

[35] Y. Li, W. gang Li, Y. tao Zhao, and A. Liu, "Opposition-based multi-objective whale optimization algorithm with multi-leader guiding," *Soft Computing*, vol. 25, no. 24, pp. 15131–15161, Dec. 2021, doi: 10.1007/s00500-021-06390-0.

[36] M. A. El Aziz, A. A. Ewees, and A. E. Hassanien, "Multi-objective whale optimization algorithm for content-based image retrieval," *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 26135–26172, Oct. 2018, doi: 10.1007/s11042-018-5840-9.

[37] M. Li, X. Yu, B. Fu, and X. Wang, "A modified whale optimization algorithm with multi-strategy mechanism for global optimization problems," *Neural Computing and Applications*, vol. 35, no. 12, pp. 8967–8986, Dec. 2023, doi: 10.1007/s00521-023-08287-5.

[38] R. Oma, S. Nakamura, T. Enokido, and M. Takizawa, "An energy-efficient model of fog and device nodes in IoT," in *Proceedings - 32nd IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2018*, pp. 301–306, 2018, doi: 10.1109/WAINA.2018.00102.

[39] F. Alenizi and O. Rana, "Dynamically controlling offloading thresholds in fog systems," *sensors*, vol. 21, no. 7, p. 2512, 2021.

[40] F. S. Abkenar, Y. Zeng, and A. Jamalipour, "Energy Consumption Tradeoff for Association-Free Fog-IoT," in *IEEE International Conference on Communications*, pp. 1–6, 2019, doi: 10.1109/ICC.2019.8761255.

[41] F. S. Abkenar and A. Jamalipour, "Energy Optimization in Association-Free Fog-IoT Networks," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 2, pp. 404–412, Jun. 2020, doi: 10.1109/TGCN.2019.2962540.

[42] B. Dhanalakshmi, L. SaiRamesh, and K. Selvakumar, "Intelligent energy-aware and secured QoS routing protocol with dynamic mobility estimation for wireless sensor networks," *Wireless Networks*, vol. 27, no. 2, pp. 1503–1514, Mar. 2021, doi: 10.1007/s11276-020-02532-8.

[43] T. M. Behera *et al.*, "Energy-Efficient Routing Protocols for Wireless Sensor Networks: Architectures, Strategies, and Performance," *Electronics (Switzerland)*, vol. 11, no. 15, p. 2282, Jul. 2022, doi: 10.3390/electronics11152282.

[44] P. SumanPrakash *et al.*, "Learning-driven Continuous Diagnostics and Mitigation program for secure edge management through Zero-Trust Architecture," *Computer Communications*, vol. 220, pp. 94–107, 2024, doi: 10.1016/j.comcom.2024.04.007.

[45] M. Bhavsingh, V. K. Uppalapati, G. S. Rao, L. Addepalli, S. V. Sagar, and J. L. Mauri, "Enhancing Airway Assessment with a Secure Hybrid Network-Blockchain System for CT & CBCT Image Evaluation," *International Research Journal of Multidisciplinary Technovation*, vol. 6, no. 2, pp. 51–69, 2024, doi: 10.54392/irjmt2425.

[46] V. S. Saranya, G. Subbarao, D. Balakotaiah, M. Bhavsingh, K. S. Babu, and S. R. Dhanikonda, "Real-Time Traffic Flow Optimization using Adaptive IoT and Data Analytics: A Novel DeepStreamNet Model," *4th International Conference on Sustainable Expert Systems, ICSES*

2024 - Proceedings, vol. 15, no. 10, pp. 312–320, 2024, doi: 10.1109/ICSES63445.2024.10763109.

[47] K. V. K. Reddy, C. M. Rao, M. Archana, Z. Begum, M. Bhavsingh, and H. Ravikumar, "VisiDriveNet: A Deep Learning Model for Enhanced Autonomous Navigation in Urban Environments," in *2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 1294–1300, Oct. 2024, doi: 10.1109/i-smac61858.2024.10714627.

[48] K. Dasari, M. A. Ali, N. B. Shankara, K. D. Reddy, M. Bhavsingh, and K. Samunnisa, "A Novel IoT-Driven Model for Real-Time Urban Wildlife Health and Safety Monitoring in Smart Cities," in *8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2024 - Proceedings*, pp. 122–129, Oct. 2024, doi: 10.1109/I-SMAC61858.2024.10714601.

[49] K. V. Ramana, G. H. K. Yadav, P. H. Basha, L. V. Sambasivarao, Y. V. B. K. Rao, and M. Bhavsingh, "Secure and Efficient Energy Trading using Homomorphic Encryption on the Green Trade Platform," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 1s, pp. 345–360, Sep. 2024.

[50] P. Kumar, M. K. Gupta, C. R. S. Rao, M. Bhavsingh, and M. Srilakshmi, "A Comparative Analysis of Collaborative Filtering Similarity Measurements for Recommendation Systems," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 3s, pp. 184–192, Mar. 2023, doi: 10.17762/ijritcc.v11i3s.6180.

[51] M. S. Lakshmi, G. Rajavikram, V. Dattatreya, B. S. Jyothi, S. Patil, and M. Bhavsingh, "Evaluating the Isolation Forest Method for Anomaly Detection in Software-Defined Networking Security," *Journal of Electrical Systems*, vol. 19, no. 4, pp. 279–297, 2023, doi: 10.52783/jes.639.

[52] E. V. N. Jyothi *et al.*, "A Graph Neural Network-based Traffic Flow Prediction System with Enhanced Accuracy and Urban Efficiency," *Journal of Electrical Systems*, vol. 19, no. 4, pp. 336–349, 2023, doi: 10.52783/jes.642.

[53] S. L. Kuna and A. V. K. Prasad, "Deep Learning Empowered Diabetic Retinopathy Detection and Classification using Retinal Fundus Images," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 2s, pp. 117–127, Jan. 2023, doi: 10.17762/ijritcc.v11i2s.6045.

[54] M. Jahir Pasha, M. Pingili, K. Sreenivasulu, M. Bhavsingh, S. I. Saheb, and A. Saleh, "Bug2 algorithm-based data fusion using mobile element for IoT-enabled wireless sensor networks," *Measurement: Sensors*, vol. 24, p. 100548, Dec. 2022, doi: 10.1016/j.measen.2022.100548.

[55] G. Ravikumar, Z. Begum, A. S. Kumar, V. Kiranmai, M. Bhavsingh, and O. K. Kumar, "Cloud Host Selection using Iterative Particle-Swarm Optimization for Dynamic Container Consolidation," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 10, no. 1, pp. 247–253, Dec. 2022, doi: 10.17762/ijritcc.v10i1s.5846.

[56] G. Yedukondalu, K. Samunnisa, M. Bhavsingh, I. S. Raghuram, and A. Lavanya, "MOCF: A Multi-Objective Clustering Framework using an Improved Particle Swarm Optimization Algorithm," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 10, no. 10, pp. 143–154, Oct. 2022, doi: 10.17762/ijritcc.v10i10.5743.

[57] M. S. Lakshmi, K. S. Ramana, M. J. Pasha, K. Lakshmi, N. Parashuram, and M. Bhavsingh, "Minimizing the Localization Error in Wireless Sensor Networks Using Multi-Objective Optimization Techniques," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 10, no. 2s, pp. 306–312, Dec. 2022, doi: 10.17762/ijritcc.v10i2s.5948.

[58] P. S. Prakash, M. Janardhan, K. Sreenivasulu, S. I. Saheb, S. Neeha, and M. Bhavsingh, "Mixed Linear Programming for Charging Vehicle Scheduling in Large-Scale Rechargeable WSNs," *Journal of Sensors*, vol. 2022, pp. 1–13, Sep. 2022, doi: 10.1155/2022/8373343.

[59] B. Swathi, S. Veerabomma, M. Archana, D. Bhadru, N. L. Somu, and M. Bhavsingh, "Edge-Centric IoT Health Monitoring: Optimizing Real-Time Responsiveness, Data Privacy, and Energy Efficiency," in *2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*, pp. 354–361, Jan. 2025, doi: 10.1109/icmcsi64620.2025.10883456.

[60] L. Chan, K. Gomez Chavez, H. Rudolph, and A. Hourani, "Hierarchical routing protocols for wireless sensor network: a compressive survey," *Wireless Networks*, vol. 26, no. 5, pp. 3291–3314, Jan. 2020, doi: 10.1007/s11276-020-02260-z.

[61] K. Guleria and A. K. Verma, "Comprehensive review for energy efficient hierarchical routing protocols on wireless sensor networks," *Wireless Networks*, vol. 25, no. 3, pp. 1159–1183, Mar. 2019, doi: 10.1007/s11276-018-1696-1.

[62] M. A. Jubair *et al.*, "Bat optimized link state routing protocol for energy-aware mobile ad-hoc networks," *Symmetry*, vol. 11, no. 11, p. 1409, Nov. 2019, doi: 10.3390/sym11111409.

[63] R. S. Abujassar, "A multi path routing protocol with efficient energy consumption in IoT applications real time traffic," *Eurasip Journal on Wireless Communications and Networking*, vol. 2024, no. 1, Jun. 2024, doi: 10.1186/s13638-024-02377-1.

[64] R. Dass *et al.*, "A Cluster-Based Energy-Efficient Secure Optimal Path-Routing Protocol for Wireless Body-Area Sensor Networks," *Sensors*, vol. 23, no. 14, p. 6274, Jul. 2023, doi: 10.3390/s23146274.

[65] N. Harini, D. S. N. Srivalli, A. Asritha, A. Govardhini, G. Pujitha, and B. G. Bhavani, "CyberSecurity Intrusion Detection in Industry 4.0 WSN's Using ML/DL," *International Journal of Computer Engineering in Research Trends*, vol. 12, no. 2, pp. 18–28, Feb. 2025, doi: 10.22362/ijcert/2025/v12/i2/v12i202.

[66] P. Ghosh, A. Losalka, and M. J. Black, "Resisting Adversarial Attacks Using Gaussian Mixture Variational Autoencoders," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 541–548, Jul. 2019, doi: 10.1609/aaai.v33i01.3301541.

[67] J. R. Choudhury, A. Saha, S. Roy, and S. Dutta, "Robust Classification of High-Dimensional Data Using Data-Adaptive Energy Distance," *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 86-101, Jun. 2023, doi: 10.1007/978-3-031-43424-2_6.

[68] Y. Tang, "Deep Learning using Linear Support Vector Machines," Jun. 2013. doi: 10.48550/arXiv.1306.0239.

[69] S. Demir and E. K. Sahin, "Application of state-of-the-art machine learning algorithms for slope stability prediction by handling outliers of the dataset," *Earth Science Informatics*, vol. 16, no. 3, pp. 2497–2509, Jul. 2023, doi: 10.1007/s12145-023-01059-8.

[70] M. Singh and L. Shrivastava, "Multi-objective optimized multi-path and multi-hop routing based on hybrid optimization algorithm in wireless sensor networks," *Wireless Networks*, vol. 30, no. 4, pp. 2715–2731, Mar. 2024, doi: 10.1007/s11276-024-03686-5.

[71] S. Demir and E. K. Sahin, "Application of state-of-the-art machine learning algorithms for slope stability prediction by handling outliers of the dataset," *Earth Science Informatics*, vol. 16, no. 3, pp. 2497–2509, Jul. 2023, doi: 10.1007/s12145-023-01059-8.

[72] M. Li, G. hui Xu, L. Zeng, and Q. Lai, "Hybrid whale optimization algorithm based on symbiosis strategy for global optimization," *Applied Intelligence*, vol. 53, no. 13, pp. 16663–16705, Dec. 2023, doi: 10.1007/s10489-022-04132-9.

[73] M. H. Nadimi-Shahraki, H. Zamani, Z. Asghari Varzaneh, and S. Mirjalili, "A Systematic Review of the Whale Optimization Algorithm: Theoretical Foundation, Improvements, and Hybridizations," *Archives of Computational Methods in Engineering*, vol. 30, no. 7, pp. 4113–4159, Mar. 2023, doi: 10.1007/s11831-023-09928-7.

[74] S. Demir and E. K. Sahin, "Application of state-of-the-art machine learning algorithms for slope stability prediction by handling outliers of the dataset," *Earth Science Informatics*, vol. 16, no. 3, pp. 2497–2509, Jul. 2023, doi: 10.1007/s12145-023-01059-8.

[75] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[76] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053–1073, Feb. 2016, doi: 10.1007/s00521-015-1920-1.

[77] E. Petrova and A. El-Sayed, "Multi-objective Optimization for Link Stability in IoT-Fog-Cloud Architectures," *International Journal of Computer Engineering in Research Trends*, vol. 11, no. 10, pp. 13–23, Oct. 2024, doi: 10.22362/ijcert/2024/v11/i10/v11i1002.

[78] Z. A. Khan and I. A. Aziz, "Ripple-Induced Whale Optimization Algorithm for Independent Tasks Scheduling on Fog Computing," *IEEE Access*, vol. 12, pp. 65736–65753, 2024, doi: 10.1109/ACCESS.2024.3398017.

[79] J. Jiang, Z. Li, Y. Tian, and N. Al-Nabhan, "A Review of Techniques and Methods for IoT Applications in Collaborative Cloud-Fog Environment," *Security and Communication Networks*, vol. 2020, 2020, doi: 10.1155/2020/8849181.

[80] S. Ijaz, S. G. Ahmad, K. Ayyub, E. U. Munir, and N. Ramzan, "Energy-efficient time and cost constraint scheduling algorithm using improved multi-objective differential evolution in fog computing," *Journal of Supercomputing*, vol. 81, no. 1, Nov. 2025, doi: 10.1007/s11227-024-06550-7.

[81] M. Yashwanth, M. G. Shukur, and D. M. R, "A Hybrid Cloud-Based Predictive Analytics Framework: Balancing Scalability, Cost Efficiency, and Data Security in Big Data Processing," *International Journal of Computer Engineering in Research Trends*, vol. 11, no. 6, pp. 12–21, Jun. 2024, doi: 10.22362/ijcert/2024/v11/i6/v11i602%20.

[82] T. P. Raptis, A. Passarella, and M. Conti, "Performance analysis of latency-aware data management in industrial IoT networks," *Sensors (Switzerland)*, vol. 18, no. 8, p. 2611, 2018, doi: 10.3390/s18082611.

[83] P. Maiti, B. Sahoo, and A. K. Turuk, "Low Latency Aware Fog Nodes Placement in Internet of Things Service Infrastructure," *Journal of Circuits, Systems and Computers*, vol. 31, no. 1, p. 2250017, 2022, doi: 10.1142/S0218126622500177.

[84] S. Suresh and N. Kannan, "Direct adaptive neural flight control system for an unstable unmanned aircraft," *Applied Soft Computing Journal*, vol. 8, no. 2, pp. 937–948, 2008, doi: 10.1016/j.asoc.2007.07.009.

[85] G. Wu *et al.*, "MECCAS: Collaborative Storage Algorithm Based on Alternating Direction Method of Multipliers on Mobile Edge Cloud," in *Proceedings - 2017 IEEE 1st International Conference on Edge Computing, EDGE 2017*, pp. 40–46, 2017, doi: 10.1109/IEEE.EDGE.2017.14.

[86] M. O'Connor, Y. Tanaka, and MallaReddy, "Optimizing IoT-Fog-Cloud Networks: A Machine Learning and Optimization Approach," *Macaw International Journal of Advanced Research in Computer Science and Engineering*, vol. 10, no. 2, pp. 1–10, Sep. 2024, doi: 10.70162/mijarcse/2024/v10/i2/v10i201.

[87] T. P. Raptis, A. Passarella, and M. Conti, "Maximizing industrial IoT network lifetime under latency constraints through edge data distribution," *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, pp. 708-713, 2018, doi: 10.1109/ICPHYS.2018.8390794.

[88] M. González, L. Svensson, and Bhavsingh, "Adaptive Resource Management in IoT-Fog-Cloud Networks via Hybrid Machine Learning Models," *International Journal of Computer Engineering in Research Trends*, vol. 11, no. 8, pp. 1–11, Aug. 2024, doi: 10.22362/ijcert/2024/v11/i8/v11i801.

[89] F. Y. Alghayadh, S. R. Jena, D. Gupta, S. Singh, I. B. Bakhriddinovich, and Y. Batla, "Dynamic data-driven resource allocation for NB-IoT performance in mobile devices," *International Journal of Data Science and Analytics*, vol. 9, no. 1, pp. 1–15, Feb. 2024, doi: 10.1007/s41060-023-00504-7.

[90] O. Ivanova and M. Al-Hassan, "Link Stability Enhancement in IoT-Fog-Cloud Systems via Advanced Optimization," *Frontiers in Collaborative Research*, vol. 2, no. 2, pp. 25–36, Jun. 2024, doi: 10.70162/fcr/2024/v2/i2/v2i203.