Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

BioTrust-QoS: A Bio-Inspired, QoS-Aware, and Secure Routing Protocol for IoT-Based Mobile Ad Hoc Networks

Shajin Nargunam A , Jebin Bose S , Samuel Blessed Nayagam P V
Noorul Islam Centre for Higher Education

Abstract - Mobile Ad Hoc Networks (MANETs) composed of Internet of Things (IoT) devices introduce a unique set of challenges for designing effective routing protocols. Imagine a city where vehicles—representing IoT devices—move unpredictably without traffic lights or fixed roads. In such an environment, ensuring every vehicle reaches its destination safely and efficiently becomes a complex task. Similarly, MANETs operate without fixed infrastructure, face frequent changes as nodes move, must cope with limited battery power, and are exposed to persistent security risks. Traditional routing protocols, such as the Ad hoc On-Demand Distance Vector (AODV), do not address these needs: they lack mechanisms to assess trustworthiness, fail to consider quality- of-service (QoS) requirements like bandwidth and latency, and only attempt to find new routes after a disruption has already occurred.

The proposed BioTrust-QoS protocol addresses these challenges by weaving together several advanced technologies—much like a well-orchestrated team, where each player brings a unique skill set to solve complex problems. BioTrust-QoS integrates bio-inspired optimization, multidimensional trust evaluation, QoS awareness, and machine-learning-based mobility prediction. The protocol employs Modified Proportional Topology Optimization (MPTO) to group nodes into resilient clusters, akin to forming neighborhoods for better coordination and efficiency. The selection of cluster heads alternates between two algorithms: Enhanced Seeker Search Optimization (ESSO) and a trust-focused variant of the Flower Pollination Algorithm (M-TCFPA), depending on the network's state—similar to how a team captain is chosen based on current game strategy. Trustworthiness is assessed across three domains—QoS, social behavior, and energy—using an adaptive framework that shifts its emphasis according to the level of environmental risk, much like adjusting security checks at an airport during heightened alerts. When discovering routes, BioTrust- QoS evaluates a composite cost that simultaneously accounts for delay, available bandwidth, residual energy, and trust, ensuring a holistic approach to route selection. To adapt to constantly changing conditions, the protocol leverages opportunistic forwarding, allowing data to take advantage of the most favorable paths as they become available. Furthermore, a deep recurrent neural network (ML-DRNN) anticipates node movements and proactively repairs routes, resulting in approximately a 75% reduction in end-to-end delay compared to AODV. To maintain robust security, BioTrust-QoS incorporates an Intelligent Dynamic Trust (IDT) model that blends Beta reputation systems with a hybrid Firefly-Whale optimisation algorithm to counter evolving attacks. Extensive evaluations using the NS-3 simulator, across varying network sizes (50 to 250 nodes) and under security threats from 2 to 10 malicious attackers, demonstrate significant improvements: packet delivery increases by over 41%, end-to-end delay drops by about 76%, and the protocol successfully identifies more than 85% of black hole and selfish nodes, while keeping false alarms low. These substantial gains highlight the effectiveness of combining optimisation, prediction, and adaptive trust—capabilities notably absent in conventional protocols.

Index Terms - Mobile Ad Hoc Networks (MANETs), Internet of Things (IoT), routing protocols, bio-inspired optimization, trust management, quality of service (QoS), security, cluster head selection, opportunistic routing.

I. INTRODUCTION

The Internet of Things (IoT) is now woven into the fabric of daily life, with billions of devices across the globe constantly sensing, collecting, and acting on data. Consider scenarios ranging from smart hospitals filled with connected medical devices, to fleets of rescue vehicles coordinating in real time during emergencies, to precision agriculture in remote farmlands. In many of these environments, reliable internet infrastructure is absent or disrupted—such as in disaster-stricken regions, military operations on the move, wildlife monitoring outposts, or isolated agricultural fields. Here, IoT devices must collaborate directly, forming Mobile Ad Hoc Networks (MANETs). Unlike traditional networks, MANETs have no central control; instead, each device takes on

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

the dual responsibility of managing its own communication and helping to relay messages for others—much like a bucket brigade passing water hand-to-hand in a coordinated effort to put out a fire.

Designing effective routing strategies for IoT-based MANETs is much like organizing foot traffic in a busy open-air festival where pathways are constantly shifting and there are no permanent signposts. As devices (nodes) move, network connections change rapidly, causing well-established routes to become obsolete in moments. Since each device typically operates on battery power, preserving energy is critical—like rationing supplies on a long hiking trip. Wireless links can be unreliable, leading to lost packets in the same way messages get lost in a game of telephone played across a windy park. Security adds another layer of complexity: malicious actors might mislead the network by suggesting false paths, intentionally dropping information, or overwhelming the system with fake requests. Without a central authority to enforce rules, traditional security solutions fall short, leaving the network vulnerable to such threats.

Despite years of research, an all-encompassing routing protocol for IoT-based MANETs remains elusive. The landscape of solutions is as varied as navigation apps: proactive protocols, such as OLSR, continuously update routes for every possible

destination, much like an app that constantly recalculates your drive even when you are stuck in traffic—leading to unnecessary background chatter. Reactive protocols, like AODV, only search for new directions when required, which saves resources but can leave travelers stranded if a road unexpectedly closes. Some protocols prioritize security, akin to apps that warn users of hazards but ignore travel time and convenience (Quality of Service, or QoS). Bio-inspired algorithms are good at picking team leaders for specific tasks but cannot always predict sudden detours. Meanwhile, QoS-focused approaches can maintain smooth traffic flow but operate under the assumption that conditions are always stable. Ultimately, most current solutions only tackle part of the overall challenge, leaving important gaps unaddressed.

To bridge these gaps, this work proposes BioTrust-QoS: a unifying protocol that brings together bio-inspired optimization, adaptive trust management, Quality of Service awareness, and machine learning-driven mobility prediction. Think of BioTrust-QoS as a next-generation navigation system—one that not only suggests the fastest route, but also verifies the reliability of each path, forecasts traffic disruptions before they occur, and flexibly shifts strategies as conditions evolve. This synergy is achieved through six seamlessly integrated components.

This paper contends that truly effective routing in IoT-based MANETs must adopt a holistic strategy—one that interlaces bio-inspired optimization, layered trust management, robust QoS mechanisms, and foresight from machine learning within a unified, coherent framework. BioTrust-QoS embodies this vision, delivering its capabilities through six interconnected functional layers, each addressing a crucial aspect of real-world network requirements.

A. Motivation and Key Challenges

This work is motivated by real-world observations. Deployments of IoT-MANETs in disaster zones, military operations, and farms all reveal similar issues with current protocols.

Challenge 1: Routes often break when nodes move. For example, first responders may suddenly lose connectivity in disaster areas. Most protocols react to failures, causing lost data and delays. Predictive methods are needed to fix routes before failures occur.

Challenge 2: Routing objectives sometimes conflict. Good routes should be short for low delay, energy-efficient, secure, and stable. Yet a short route may pass through risky or weak nodes. Most protocols focus on a single goal or use fixed weights that do not adapt as network needs change. For low-risk smart homes, direct observation works. Military networks need strong, multi-node trust systems to handle attacks. The level of trust checking must adapt as threats and settings change.

Challenge 3: Dynamic Trust Requirements. A trust model that works for a home automation network is insufficient for a military operation. Home networks can use lightweight direct observation because the threat model is mild. Military networks need consensus-based trust that aggregates recommendations from multiple neighbours to detect sophisticated attacks. Moreover, the required trust level can change over time as threat conditions evolve. A protocol should adapt by adjusting the complexity and factors in its trust computation method based on the current security context.

Challenge 4: Attack risk grows as the network expands. When nodes increase from 50 to 250, control traffic rises, especially if attackers flood the network. Good protocols must group traffic by clusters and share trust checks to avoid slowdowns.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Challenge 5: Real-Time QoS Guarantees. Real-time use requires strong performance guarantees. Video and audio links cannot tolerate much delay or packet loss. Average results are insufficient. Strict quality-of-service (QoS) rules are needed. Routing must verify that paths meet all requirements before use and switch quickly if they fail.

B. Proposed Solution: BioTrust-QoS

BioTrust-QoS addresses these five challenges through the following innovations.

Innovation 1: Hierarchical Clustering with MPTO. BioTrust-QoS organises nodes into clusters using MPTO, which groups devices by proximity to minimise network-wide route discovery. Intelligent cluster sizing reduces bottlenecks and localises node mobility effects, significantly lowering communication overhead.

Innovation 2: Multi-Objective Bio-Inspired Cluster Head Selection. Selecting a cluster head is key. ESSO works well with a single clear goal. M-TCFPA finds several good choices when goals compete. Both consider mobility, signal, energy, and cooperation.

Innovation 3: Context-Aware Three-Level Trust. Trust is measured in three ways. In safe places, direct observation suffices. In public or business networks, node recommendations are used. High-risk cases require specialised detection to stop attacks such as wormholes or black holes. Nodes adjust their trust method as risks change.

Innovation 4: Composite QoS-Trust Cost Function. Route selection considers delay, bandwidth, energy, and trust. Weights adjust to fit the application. Any route missing a key requirement is discarded.

Innovation 5: ML-DRNN Mobility Prediction. ML-DRNN predicts node movement using two stacked LSTM layers. If a link is predicted to fail within 2 seconds, route discovery begins immediately to maintain connectivity.

Innovation 6: Intelligent Dynamic Trust with Firefly-Whale Optimization. The trust system uses Beta trust and a hybrid Firefly-Whale algorithm. This mix helps defend against attacks and maintains stable, energy-saving routes as threats evolve.

Innovation 7: Opportunistic Forwarding with Real-Time Adaptation. Instead of relying on a fixed next hop, each node keeps a list of neighbors meeting trust and QoS criteria. Before transmission, the node selects the optimal forwarder based on real-time signal quality and trust assessments, enabling rapid adaptation without restarting route discovery. BioTrust-QoS thus accounts for all moving parts in real-world IoT MANETs.

C. Contributions

The main contributions of this paper are as follows.

1) BioTrust-QoS combines bio-inspired clustering, multidimensional trust, QoS awareness, machine learning-based mobil- ity prediction, and opportunistic forwarding. No previous protocol brings together all these elements within a single framework.
2) For clustering, a Modified Proportional Topology Optimisation (MPTO) algorithm is proposed. This algorithm parti- tions nodes based on their locations, maintains balanced clusters, reduces unnecessary signalling, and ensures network connectivity.
3) On the security front, a three-level adaptive trust model is employed. In low-risk scenarios, the model utilises lightweight direct observation, while in higher-risk situations, it escalates to more complex, consensus-based trust evaluations.
4) Cluster head selection leverages two bio-inspired methods. Enhanced Seeker Search Optimisation (ESSO) addresses single-objective tasks, while the Multi-Objective Trust-Centric Flower Pollination Algorithm (M-TCFPA) is applied when multiple objectives and trade-offs must be balanced.
5) We designed a cost function that balances delay, bandwidth, energy, and trust while ensuring hard QoS requirements are always met.
6) For mobility prediction, we built a Multi-Layer Deep Recurrent Neural Network (ML-DRNN) that predicts node move- ment, allowing route fixes before links break. Tests showed it cut end-to-end delay by 75% compared to AODV.
7) To provide long-term defence against attacks, we created the Intelligent Dynamic Trust (IDT) model, which combines Beta reputation with a hybrid Firefly-Whale optimisation. This combo makes the network tough against evolving threat.
8) We tested BioTrust-QoS in NS-3 under loaded scenarios with 50 to 250 nodes and 2 to 10 malicious nodes. Results showed BioTrust-QoS delivered more packets, responded faster, maintained higher throughput, and detected attacks more effectively than AODV, ESCT, or ETRS. ETRS.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

D. Paper Organization

As for the paper itself, Section II surveys related work in the main areas. Section III outlines the architecture of BioTrust-QoS. Section IV covers how we set up simulations and compared our approach to others. Section V wraps things up.

E. Notational Conventions

Throughout this paper, we use the following notations. The set of nodes is denoted by N with $|N| = N$. Node positions are given by $p_i(t)$ at time t. Residual energy is $E_i(t)$. Trust from node i to node j is $_{i,j}$. Clusters are C_k with centroids $_k$. The path cost function is $Cost(p)$. The trust threshold is τ_{trust}. Link bandwidth, delay, and loss rate are B_{uv}, δ_{uv}, and l_{uv} respectively. The ML-DRNN predicts future positions $\hat{p}_i(t + \Delta t)$. The Beta reputation parameters are α_j (successes) and β_j (failures).

II. RELATED WORK

Secure, energy-efficient, and QoS-aware routing protocols are critical for Mobile Ad Hoc Networks (MANETs) and IoT- based MANETs, especially as these networks operate without centralized control. Over the past two decades, researchers have applied bio-inspired optimization, clustering, trust management, and hybrid cryptographic methods to address these challenges. Here, we review key contributions and clarify how our approach builds on and differs from existing work.

A. Bio-Inspired and Meta-Heuristic Optimisation for Routing

Bio-inspired algorithms are commonly used for routing and cluster head selection in MANETs. These methods handle multiple objectives and adapt to changing network conditions without requiring full network knowledge.

Kocherla et al. [1] proposed an energy efficiency enhancement scheme for prolonging network lifetime in multi-conditional multi-sensor based wireless sensor networks. Their work demonstrated that careful optimization of energy consumption can significantly extend network operational lifetime. Similarly, Venkatasubramanian et al. [2] introduced coral reef optimization for cluster head selection and optimal multipath detection in MANET environments. Their approach showed that biologically inspired optimization can effectively balance multiple conflicting objectives in dynamic network conditions.

Suresh Kumar et al. [3] developed a cluster head selection and energy efficient multicast routing protocol for MANETs. Their work focused on optimal route selection while considering energy constraints. Boddu et al. [4] presented a novel geo-routing potency based optimum spider monkey approach for avoiding congestion in energy efficient mobile ad hoc networks. The spider monkey optimization technique proved effective in maintaining network performance under congestion.

Tawfeeq[6] used a connectivity probability method based on Poisson distribution and residual energy for cluster head selection in MANETs, providing a rigorous statistical framework. Sindhanaiselvan et al.[7] developed a dynamic topology method for cluster head selection, highlighting the need for adaptive clustering in mobile networks.

Jubair et al.[8] proposed a cluster head selection and hybrid cryptography routing protocol for VANETs, combining security with quality of service. This approach targets vehicular networks rather than general MANETs.

Vatambeti et al.[9] combined grey wolf and dragonfly algorithms for routing optimization in wireless sensor networks, showing that hybrid bio-inspired methods can outperform single algorithms. Hamza and Vigila[33] used emperor penguin optimization with a fuzzy genetic algorithm for energy-efficient cluster head selection, demonstrating that fuzzy logic helps manage uncertain network data.

Mahadevachar and Hosur[34] developed an energy-efficient routing protocol using battle royale optimization as a meta-heuristic. Abdelhamid et al.[36] introduced the waterwheel plant algorithm as a new metaheuristic for routing optimization.

B. Trust-Based and Secure Routing Protocols

MANETs require decentralized trust models because centralized authentication is not practical in infrastructure-less settings. Several approaches have been proposed to address this challenge.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Vigenesh and Santhosh [5] introduced a stream region sink position analysis model to detect routing attacks in MANETs, using behavioral analysis to identify malicious nodes instead of relying solely on cryptography. Vatambeti and Mamidisetti [35] applied ensemble deep learning models for routing attack detection in IIoT, showing that machine learning can address security challenges in these networks.

Veeraiah et al. [26] proposed a trust-aware, secure, and energy-efficient hybrid protocol for MANETs, combining trust evaluation with energy considerations. This approach informed our own multi-dimensional trust assessment. Bondada et al. [15] focused on data security-based routing using key management, highlighting the need for effective cryptographic key distribution in ad hoc networks.

Saravanan et al. [13] developed a trust-aware ad hoc routing protocol that integrates key management and energy-efficient cluster head selection. While their method combines trust, security, and energy optimization, it does not address mobility prediction or QoS constraints, which are central to our approach.

Mukhedkar and Kolekar [18] used a hybrid optimization algorithm for trust-based secure routing. Devi et al. [21] designed a secure cross-layer routing protocol with an authentication key management scheme. Ninu [22] proposed an intrusion detection system that applies exponential Henry gas solubility optimization with a deep neuro-fuzzy network.

Desai and Jhaveri [23] investigated secure routing with predictive methods, showing that anticipating network behavior can be more effective than reactive strategies. Aluvala and Rajasekhar [30] combined adaptive cuckoo search with entropy-based signature authentication for secure routing.

C. QoS-Aware and Energy-Efficient Routing

Incorporating QoS requirements into routing is essential for supporting multimedia and real-time IoT applications. Research in this area has produced a range of approaches aimed at improving network performance under dynamic conditions.

Shajin Nargunam and Sebastian [31], [32] introduced a cluster-based QoS-aware multicast routing protocol (CQMRP) for mobile ad hoc networks. Their results showed that clustering can maintain QoS support while adapting to topology changes. Compared to related protocols, CQMRP reduced control overhead and achieved higher packet delivery ratios, particularly in high-mobility scenarios.

Rajathi [16] advanced energy-efficient clustering by introducing a cluster coordinator-based mechanism for cluster head election. Hai et al. [17] addressed security in cloud-MANETs by combining multi-path routing with cloud computing concepts to enhance secure data transmission.

Kumaresan et al. [19] applied fuzzy marine white shark optimization to improve routing efficiency and extend network lifetime. Sharma and Saxena [20] focused on energy-efficient service discovery in MANETs, while Farheen and Jain [24] enhanced multi-path routing through hybrid modeling and optimization.

Sahu et al. [25] developed ZBLE, a zone-based multipath protocol designed to improve energy efficiency in MANET routing. Muneeswari and Manikandan [27] leveraged reinforcement learning for energy-efficient clustering and secure routing in three-dimensional networks. Kumar et al. [28] combined k-means clustering with AODV to further enhance energy efficiency. Lavanya et al. [29] optimized multipath routing by incorporating mobility metrics into link state routing. Sundaram and Ravi [12] demonstrated the effectiveness of bio-inspired optimization, specifically gray wolf optimization, for geographic routing challenges.

Srilakshmi et al. [11] unified security and routing optimization in a single framework, proposing a secure optimization routing algorithm for MANETs.

D. Research Gaps and Our Contributions

Most existing protocols address trust, QoS, and energy efficiency as separate objectives or optimize them one after another. BioTrust-QoS instead optimizes all three at once using a composite cost function that adapts to application requirements.

Most bio-inspired protocols do not use machine learning for mobility prediction. Those that attempt prediction often rely on linear extrapolation, which is unreliable in highly dynamic networks. The ML-DRNN model instead learns mobility patterns from historical data and enables proactive route repairs before links fail.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Most trust models are static or slow to adapt. The context-aware three-level trust model adjusts automatically to the security environment, using direct observation under low threat and switching to consensus-based assessment as threats increase.

Few protocols combine opportunistic routing with both trust and QoS constraints in a unified framework. The candidate forwarder selection mechanism uses real-time link quality together with historical trust.

Most existing protocols use a single bio-inspired optimization algorithm. The hybrid approach combines Firefly and Whale algorithms within a dynamic trust model, improving the ability to escape local optima.

Table I summarizes the main differences between BioTrust-QoS and related protocols. BioTrust-QoS is one of the few protocols to combine bio-inspired optimization, multi-dimensional trust, QoS awareness, machine learning-based prediction, and opportunistic forwarding in a single framework.

TABLE I: Comparison with Related Work

Protocol	Bio-Inspired	Trust	QoS	ML Prediction	Opportunistic
Kocherla et al. [1]	Yes	No	Partial	No	No
Venkatasubramanian et al. [2]	Yes (Coral Reef)	No	No	No	No
Suresh Kumar et al. [3]	Yes	No	Yes	No	No
Boddu et al. [4]	Yes (Spider Monkey)	No	Partial	No	No
Tawfeeq [6]	No	No	No	No	No
Jubair et al. [8]	Yes	Yes (Hybrid Crypto)	Yes	No	No
Vatambeti et al. [9]	Yes (Grey Wolf + Dragonfly)	No	Yes	No	No
Saravanan et al. [13]	Yes (PSO)	Yes	Partial	No	No
Veeraiah et al. [26]	No	Yes	Yes	No	No
Nargunam and Sebastian [32]	No	No	Yes	No	No
Srilakshmi et al. [11]	Yes	Yes	Partial	No	No
BioTrust-QoS (Proposed)	**Yes (ESSO/M-TCFPA)**	**Yes**	**Yes**	**Yes (ML-DRNN)**	**Yes**

III. Architectural Design and Methodology

A. System Overview

We introduced BioTrust-QoS, a routing protocol for IoT-driven Mobile Ad Hoc Networks (MANETs) that is bio-inspired, QoS-focused, and secure. The name reflects biological optimization, service quality, and built-in security. After testing other options, we chose a six-plane architecture that clarifies each part's role while allowing communication between planes when needed.

The six functional planes are:

1) Network Organization Plane – builds stable clusters using a Modified Proportional Topology Optimization (MPTO) algorithm.
2) Trust Assessment Plane – computes multi-dimensional, context-aware trust scores for every node.
3) Bio-inspired Service Node Selection Plane – elects cluster heads (CHs) using either Enhanced Seeker Search Optimization (ESSO) or a multi-objective Flower Pollination Algorithm (M-TCFPA).
4) QoS-Aware Secure Route Discovery Plane – performs on-demand routing with a composite cost function and opportunistic forwarding.
5) AI-Enhanced Path Optimization Plane – predicts node mobility with a Multi-Layer Deep Recurrent Neural Network (ML-DRNN) and refines routes using Le´vy flight search.
6) Security Enforcement & Adaptation Plane – runs a lightweight intrusion detection system (IDS), an Intelligent Dynamic Trust (IDT) model, and a hybrid Firefly–Whale optimization.

All these planes communicate via a shared control channel, and each node maintains three small data structures: a neighbor table (for link quality and trust), a routing cache (only paths that pass QoS checks), and a mobility history buffer (that feeds the ML-DRNN predictor). We kept these tables small, so even basic IoT devices can handle them.

B. Hierarchical Network Organization Using MPTO

Tests with flat routing showed control overhead spiraled as the network grew. Clustering helped control this. To handle rapidly changing topologies and high mobility, we introduced the Modified Proportional Topology Optimization (MPTO) algorithm. Unlike classic clustering like LEACH, MPTO balances clusters by considering both physical spread and traffic.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

1) Problem Formulation: Let the set of IoT nodes be N = {1, 2, . . . , N } with positions p_i (t) and residual energy E_i (t). TThe clustering goal: keep clusters well-balanced and close-knit (with short distances within clusters).

$$\min \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} \|\mathbf{p}_i - \mathbf{c}_k\|^2 + \lambda \cdot \mathrm{Var}\big(|\mathcal{C}_1|, \ldots, |\mathcal{C}_K|\big)$$

where c_k is the centroid of cluster C_k, K is the number of clusters, and λ is a balancing coefficient that we tuned experimentally to 0.3. If λ is too small, clusters become highly unbalanced; if too large, the clustering ignores the actual topology.

2) MPTO Algorithm Steps: MPTO works as follows. First, we pick K initial centroids by choosing the farthest nodes from each other (not randomly, to avoid poor results). Each node joins the cluster with the nearest centroid. Then, we check cluster sizes: if a cluster is too large, we split it; if too small, we merge it with the closest neighbor. Sensitivity analysis showed theta = 0.25 is a good threshold. Centroids are updated, and the process repeats until stable or the maximum loops are reached.

Algorithm 1 Modified Proportional Topology Optimization (MPTO)

Require: Node set $\mathcal{N}$, positions $\mathbf{p}_i$, number of clusters K, balance factor λ, tolerance θ
Ensure: Clusters $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ and candidate cluster heads
  Initialize K centroids using farthest-first traversal
  repeat
    for each node $i \in \mathcal{N}$ **do**
      Assign i to cluster $\mathcal{C}_k$ with smallest $\|\mathbf{p}_i - \mathbf{c}_k\|^2$
    end for
    for each cluster $\mathcal{C}_k$ **do**
      if $|\mathcal{C}_k| > (1+\theta) \cdot N/K$ **then**
        Split $\mathcal{C}_k$ into two clusters
      else if $|\mathcal{C}_k| < (1-\theta) \cdot N/K$ **then**
        Merge $\mathcal{C}_k$ with its nearest neighbor cluster
      end if
    end for
    for each cluster $\mathcal{C}_k$ **do**
      Update centroid $\mathbf{c}_k = \frac{1}{|\mathcal{C}_k|} \sum_{i \in \mathcal{C}_k} \mathbf{p}_i$
    end for
  until convergence or max iterations reached
  Output: Clusters and nodes closest to centroids as temporary CH candidates

Algorithm 1 Modified Proportional Topology Optimization (MPTO)

Require: Node set $\mathcal{N}$, positions $\mathbf{p}_i$, number of clusters K, balance factor λ, tolerance θ
Ensure: Clusters $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ and candidate cluster heads
  Initialize K centroids using farthest-first traversal
  repeat
    for each node $i \in \mathcal{N}$ **do**
      Assign i to cluster $\mathcal{C}_k$ with smallest $\|\mathbf{p}_i - \mathbf{c}_k\|^2$
    end for
    for each cluster $\mathcal{C}_k$ **do**
      if $|\mathcal{C}_k| > (1+\theta) \cdot N/K$ **then**
        Split $\mathcal{C}_k$ into two clusters
      else if $|\mathcal{C}_k| < (1-\theta) \cdot N/K$ **then**
        Merge $\mathcal{C}_k$ with its nearest neighbor cluster
      end if
    end for
    for each cluster $\mathcal{C}_k$ **do**
      Update centroid $\mathbf{c}_k = \frac{1}{|\mathcal{C}_k|} \sum_{i \in \mathcal{C}_k} \mathbf{p}_i$
    end for
  until convergence or max iterations reached
  Output: Clusters and nodes closest to centroids as temporary CH candidates

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

The result is a set of stable clusters and a small number of cluster head candidates—nodes closest to centroids with above-average leftover energy.

C. Multi-Dimensional Trust Assessment

Trust is the backbone of our security. Without a central authority in a MANET, each node must judge its neighbors independently. Our approach combines three sources: direct observation, recommendations from other nodes, and historical behavior.

1) Trust Metrics: For a node j as observed by node i, the composite trust $T_{i,j}$ is a weighted sum:

$$T_{i,j} = w_Q \cdot T_{\text{QoS}} + w_S \cdot T_{\text{Social}} + w_E \cdot T_{\text{Energy}}$$

We define the three components as follows:

- $T_{\text{QoS}} = \frac{\text{PacketDelivered}}{\text{PacketSent}}$ this captures both competence (successful forwarding) and reliability (consistency over time).
- T_{Social} = Honesty × Intimacy × (1 − Selfishness). Honesty is measured by how often a node's reported information matches reality. Intimacy is the number of successful interactions over a sliding window. Selfishness is the fraction of forwarding opportunities that the node deliberately ignores.
- $T_{\text{Energy}} = \frac{E_{\text{residual}}}{E_{\text{initial}}}$ a node that is about to run out of battery cannot be trusted to carry out routing duties reliably.

The weights w_Q , w_S , w_E are not fixed; they change according to the deployment scenario. For a disaster-response network we set $w_E = 0.5$ to prioritize energy; for a military application we set $w_S = 0.6$ to emphasize social trust. In all cases the weights sum to one.

2) Context-Aware Trust Levels: A single trust metric wasn't enough — it's either too heavy for small devices or too limited for critical uses. So we defined three trust levels, shown in Table I.

TABLE II: Context-aware trust levels

Level	Security Context	Computation	Inputs
Low	Home / office	Direct observation only	Packet delivery ratio
Medium	Public / enterprise	Direct + neighbor recommendations	Subjective logic, majority voting
High	Military / critical	Weighted average + anomaly detection	Historical trust vectors, deviation from expected behavior

In the high-level mode, the trust update follows:

T^{new}

$$T_{i,j}^{\text{new}} = \beta \cdot T_{i,j}^{\text{direct}} + (1-\beta) \cdot \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} T_{r,j}^{\text{rec}}$$

where R is the set of recommenders. Through experimentation we found that β = 0.7 works well for military scenarios because it gives more weight to what the node directly observes while still allowing second-hand information to influence the final trust.

D. Bio-Inspired Service Node (Cluster Head) Selection

Now that MPTO has created clusters, we need to assign cluster heads to route traffic between them. We offer two bio-inspired approaches, depending on the network's needs.

1) Enhanced Seeker Search Optimization (ESSO): ESSO acts like a person seeking out the best option. Each candidate cluster head is described by four metrics: mobility, signal strength, energy use rate, and cooperation. Weights (0.3, 0.2, 0.3,

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

0.2) were chosen after trial runs. Each seeker updates its own and global best positions, and the cluster head with the top score is selected.ion Algorithm (M-TCFPA). The fitness function is:

$$F(x) = \gamma_1 \cdot (1 - \text{Mobility}(x)) + \gamma_2 \cdot \text{Signal}(x) + \gamma_3 \cdot (1 - \text{EnergyRate}(x)) + \gamma_4 \cdot \text{Cooperation}(x)$$

We chose the coefficients as $\gamma_1 = 0.3$, $\gamma_2 = 0.2$, $\gamma_3 = 0.3$, $\gamma_4 = 0.2$ after trying several combinations on a small testbed. The update rule for each seeker (candidate) is:

$$v_{id}(t+1) = v_{id}(t) + \phi_1 \cdot (p_{id} - x_{id}) + \phi_2 \cdot (g_d - x_{id})$$

Here p_{id} is the personal best position, g_d is the global best, and ϕ_1 , ϕ_2 are random numbers drawn uniformly from [0,1.5]. The node with the highest fitness becomes the cluster head.

2) Multi-Objective Flower Pollination Algorithm (M-TCFPA): For scenarios where we cannot combine multiple objectives into a single scalar (for example, when trust and energy are equally important and incommensurable), we use M-TCFPA. It generates a Pareto front of non-dominated solutions. The three objectives (all to be maximized) are:

$$f_1 = \frac{1}{|\mathcal{C}|} \sum_{j \in \mathcal{C}} T_{\text{CH},j} \qquad \text{(average trust)} \qquad (1)$$

$$f_2 = E_{\text{res}}(\text{CH}) \qquad \text{(residual energy)} \qquad (2)$$

$$f_3 = 1 \Big/ \frac{1}{|\mathcal{C}|} \sum_{j \in \mathcal{C}} \|\mathbf{p}_{\text{CH}} - \mathbf{p}_j\| \qquad \text{(inverse average distance)} \qquad (3)$$

The pollination operators are:

- **Global (biotic)**: $x_i^{t+1} = x_i^t + L(\lambda) \cdot (g_* - x_i^t)$, where $L(\lambda)$ is a Lévy flight step.
- **Local (abiotic)**: $x_i^{t+1} = x_i^t + \epsilon \cdot (x_j^t - x_k^t)$, with ϵ a small uniform random number.

After the algorithm converges, we select a solution from the Pareto front based on the application's priorities. For example, a real-time video streaming application would choose the solution with the highest f_2 (energy), while a security-critical application would choose the one with the highest f_1 .

E. QoS-Aware Secure Route Discovery

Route discovery is on-demand to avoid flooding the network with control traffic. At the same time, we don't just look for any path — it has to satisfy trust and QoS requirements from the get-go.

1) Composite Cost Function: For any path p, the cost is the sum of the normalised energy, delay, bandwidth, and trust scores of its links, each weighted equally. We normalize each metric to the range [0, 1] based on observed network extremes. Depending on the application, these weights (alpha, beta, gamma, delta) shift: more bandwidth for VoIP, more delay-tolerance for surveillance.

For a path $p = (s, n_1, n_2, \ldots, d)$, the cost is:

$$\text{Cost}(p) = \sum_{(u,v) \in p} \left[\alpha \cdot f_E(E_u) + \beta \cdot f_D(\delta_{uv}) + \gamma \cdot f_B\left(\frac{1}{B_{uv}}\right) + \delta \cdot (1 - T_{u,v}) \right]$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

The functions f_E , f_D , f_B normalise their arguments to the range [0,1] using min-max normalisation based on observed extremes in the network. The coefficients satisfy $\alpha + \beta + \gamma + \delta = 1$. For a typical surveillance application we set $\alpha = 0.2$, $\beta = 0.3$, $\gamma = 0.2$, $\delta = 0.3$. For a VoIP application we increase β to 0.5 and reduce γ to 0.1.

A route is picked only if it meets all three QoS constraints: sufficient bandwidth, delay within the cap, and loss rate below the limit. These are based on ITU-T multimedia guidelines.

$$\min_{(u,v)\in p} B_{uv} \geq B_{\text{req}} \qquad \text{(bandwidth, concave)} \qquad (4)$$

$$\sum_{(u,v)\in p} \delta_{uv} \leq D_{\max} \qquad \text{(delay, additive)} \qquad (5)$$

$$1 - \prod_{(u,v)\in p} (1 - l_{uv}) \leq L_{\max} \qquad \text{(loss rate, multiplicative)} \qquad (6)$$

We derived these constraints from the ITU-T recommendations for multimedia services.

2) Opportunistic Forwarding (ELRP-based): Fixed next hops caused problems when links suddenly failed. We switched to opportunistic forwarding. Each node keeps a list of neighbors meeting trust and QoS criteria. During transmission, the forwarder is the neighbor with the best trust-to-signal ratio, with trust weighted more (omega = 0.6) as it is more critical than signal strength.

$$\mathcal{F}_i = \{j \in \mathcal{N}(i) \mid T_{i,j} \geq \tau_{\text{trust}} \text{ and } \text{QoS}_{\text{link}(i,j)} \geq \text{QoS}_{\text{req}}\}$$

At runtime, the actual forwarder is selected as:

$$j^* = \arg\max_{j\in\mathcal{F}_i} (\omega \cdot T_{i,j} + (1-\omega) \cdot \text{SNR}_{i,j})$$

We set $\omega = 0.6$ after testing because trust is slightly more important than pure signal strength, but the latter still matters for immediate transmission quality.

3) Route Discovery Procedure: The full discovery process proceeds in four steps:

1) The source broadcasts a RREQ packet that carries the needed QoS values, the minimum trust τ_{trust} , and the running path cost.
2) Each intermediate node checks whether the path still meets the QoS constraints and whether the trust level of the previous node is above τ_{trust} . If not, the RREQ is discarded.
3) When the destination receives one or more valid RREQs, it replies with a RREP along the lowest-cost route that meets all the requirements.
4) The source saves the path in a cache with its trust level and a timestamp. The route is dropped from the cache after a timeout or if any error is detected.

F. AI-Enhanced Path Optimization and Maintenance

Even with opportunistic forwarding, mobility continues to threaten stability. To keep up, we added two more layers: path prediction and bio-inspired optimization.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

1) ML-DRNN Mobility Prediction: Each cluster head runs an ML-DRNN that takes a sequence of past positions and speeds for a node. The neural network has two stacked LSTM layers with 64 units each, dropout (0.2) to avoid overfitting, and a dense output to predict future locations. We deployed a Multi-Layer Deep Recurrent Neural Network (ML-DRNN) at each cluster head. The input to the network for node i is a sequence of its last L positions $[p_i(t - L + 1), \ldots, p_i(t)]$ together with the corresponding velocities. The architecture consists of:

- Two stacked LSTM layers with 64 units each.
- A dropout layer with rate 0.2 inserted between the LSTM layers to reduce overfitting.
- A dense output layer that predicts the future position $\hat{p}_i(t + \Delta t)$, where $\Delta t = 1$ second in our implementation.

We trained the model offline with BonnMotion traces, then fine-tuned it online by having cluster heads exchange weight updates every ten minutes. Using these predictions, nodes estimate when a link will break. If remaining time falls below two seconds, route discovery starts early. This approach noticeably reduced route breakages in tests.

$$T_{\text{exp}}(i,j) = \frac{R - \|\mathbf{p}_i - \mathbf{p}_j\|}{\|\mathbf{v}_i - \mathbf{v}_j\|}$$

If T_{exp} falls below a threshold $\tau_{exp} = 2$ seconds, the node proactively triggers a new route discovery. This preemptive action significantly reduces the number of broken routes in our simulations.

2) Le´vy Flight Path Optimization (LF-SSO-DSR): When multiple candidate routes exist, we apply a Le´vy flight search to avoid getting trapped in local optima. The fitness of a route p is defined as:

$$\Phi(p) = \frac{1}{\text{Cost}(p)} \times \text{PathTrust}(p) \times \text{EnergyEfficiency}(p)$$

where $\text{PathTrust}(p) = \min_{(u,v)\in p} T_{u,v}$ and $\text{EnergyEfficiency}(p) = \min_{(u,v)\in p}(E_u/E_{\text{initial}})$. The Lévy flight step is:

$$p_{\text{new}} = p_{\text{current}} + \alpha \cdot \text{Lévy}(\beta) \cdot (p_{\text{best}} - p_{\text{current}})$$

The step size α is set to 0.1 times the diameter of the solution space, and L´evy(β) is drawn from a distribution with $\beta = 1.5$. This search converges to the global best route in far fewer iterations than a pure genetic algorithm.

G. Security Enforcement and Dynamic Adaptation

Security is not a one-time configuration; it must adapt as the network evolves. Our design includes four mechanisms that work together.

1) Lightweight Intrusion Detection System (IDS): Every node monitors its neighbors for four types of attacks:

- Black hole: a node drops more than 90% of packets over a sliding window of 50 packets.
- Grey hole: a node drops packets selectively (e.g., only routing control packets but forwards data). We detect this by comparing the forwarding ratio for different packet types.
- Wormhole: two colluding nodes create a tunnel. We detect wormholes by observing abnormally low delays for long distances and by using packet time-to-live anomalies.
- DoS (RREQ flooding): a node sends RREQ packets at a rate higher than $\theta_{\text{flood}} = 10$ per second. This threshold was chosen based on the maximum reasonable rate in a dense MANET.

When a node is confirmed as malicious, it is added to a blacklist and quarantined for a penalty period. During quarantine, the node cannot participate in route discovery or data forwarding.

2) Intelligent Dynamic Trust (IDT) Model: Our trust system combines Beta reputation with a hybrid Firefly-Whale Optimization Algorithm (WOA). Beta reputation tracks how often each node behaves well, using Laplace smoothing so new nodes do

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

not start with zero trust. In optimization, Firefly's idea is that better solutions attract others, with attraction fading over distance (beta_0 = 1.0, eta = 0.5). Then, the Whale Optimization Algorithm uses a bubble-net approach to refine the top path. The IDT model produces routes that maximize trust and balance energy drain on nodes.The Beta reputation for node j is:

$$\mathrm{Rep}(j) = \frac{\alpha_j + 1}{\alpha_j + \beta_j + 2}$$

Here α_j counts successful interactions (packets forwarded correctly) and β_j counts failed interactions (packets dropped or altered). The +1 and +2 in the formula are Laplace smoothing factors to handle the cold-start problem.

In the Firefly part, the attractiveness between two candidate solutions i and j is:

$$\beta(r) = \beta_0 e^{-\eta r^2}$$

where r is the Euclidean distance between the solutions in the search space. We set β_0 = 1.0 and η = 0.5. The Whale Optimization part refines the best solution using the bubble-net attacking mechanism:

$$\mathbf{X}(t+1) = \mathbf{X}^*(t) - A \cdot |C \cdot \mathbf{X}^*(t) - \mathbf{X}(t)|$$

The coefficients A and C are generated as described in the original WOA paper. The IDT outputs a set of routes that maximize the Beta reputation while minimizing the variance in residual energy among the nodes on the path.

3) Dynamic Trust Threshold Adaptation: As we update the threshold dynamically based on actual attacks and false alarm rates. The learning rate was chosen after many trial runs.A lightweight ARIMA model runs every minute, forecasting trust scores for the next 10 seconds. If the forecast drops by more than 10%, we raise the threshold in advance to prevent the network from being caught off guard.

$$\tau_{\text{trust}}(t+1) = \tau_{\text{trust}}(t) + \eta \cdot \left(\text{AttackDetected}(t) - \text{FalsePositive}(t)\right)$$

The learning rate η = 0.05 was chosen empirically. In addition, we run a simple ARIMA model every 60 seconds to forecast the trust trend for the next 10 seconds. If the forecast shows a decline of more than 10%, we raise the threshold preemptively. This predictive approach reduces the window of vulnerability after an attack begins.

4) Secure Communication: All control and data packets are encrypted. Within clusters, nodes use AES-128 in CCM mode for privacy and integrity. Keys are generated between a node and its cluster head using Elliptic Curve Diffie-Hellman (ECDH) with the NIST P-256 curve. Between clusters, cluster heads use pre-issued certificates signed by a trusted authority. Keys rotate every 15 minutes or after an IDS alert to stay fresh.

H. Summary of Algorithmic Integration

The protocol as a whole acts like a finite-state machine, moving through several big stages. First up: Initialization—clusters form, and candidates for cluster head (CH) are picked using MPTO. Then, for the first 60 seconds, nodes gather direct trust info (trust bootstrapping). Next, the cluster itself holds an election using either ESSO or M-TCFPA, then chooses its final CH. After that, for routing, the source node initiates an on-demand RREQ/RREP exchange, checking costs and QoS along the way. When data needs to be forwarded, a node gets picked on the fly from a list of candidates (opportunistic forwarding). Maintenance is handled by ML-DRNN, which predicts whether a link's likely to fail soon; if so, the node quickly finds a detour. Security adaptation runs across everything: the IDS watches always, trust gets re-optimized by the IDT, and trust thresholds shift as conditions change.

1) Initialization: MPTO runs to form clusters and produce candidate CHs.
2) Trust bootstrapping: For the first T_{init} = 60 seconds, nodes use only direct observations to build initial trust values. This avoids being misled by false recommendations during the early phase.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

3) CH election: The cluster runs either ESSO or M-TCFPA on the candidate set and announces the final CH.
4) Route discovery: The source initiates an on-demand RREQ/RREP exchange with composite cost and QoS checks.
5) Opportunistic forwarding: Each data packet forwarder is selected at runtime from the candidate forwarder set.
6) Predictive maintenance: The ML-DRNN predicts future positions, and if a link is about to break, the node proactively finds a new route.
7) Security adaptation: The IDS runs continuously; the IDT model periodically re-optimizes paths; trust thresholds adjust based on events.

All parts share information via the Cross-Layer Information Base (CLIB), a compact key-value store with tables for trust, routing, mobility, and QoS. Its size remains under 50 kB per node, suitable for low-power chips like the ESP32.

I. Complexity Analysis

We took a cloWe analyzed time and memory usage (see Table II). For networks of 50 to 500 nodes, demands remain manageable. ML-DRNN predictions consume the most computation but run only at cluster heads every two seconds. Bio- inspired algorithms activate only during cluster head reelection every 5–10 minutes. In short, the protocol runs well on constrained IoT hardware.her, this gives the technical backbone for BioTrust-QoS. Next up: how we tested it, the scores it earned, and how it stacked up against other routing protocols.

For typical MANET sizes of 50 to 500 nodes, the complexities are manageable. The ML-DRNN prediction is the most expensive in terms of per-node computation, but we run it only at cluster heads (which are fewer) and only once every two seconds. The bio-inspired algorithms are executed only during cluster head election, which happens infrequently (every 5 to 10 minutes). Therefore the overall protocol is suitable for resource-constrained IoT devices.

This methodology provides the complete technical foundation for BioTrust-QoS. The next sections of the paper present the simulation setup, performance evaluation, and comparison with existing routing protocols.

TABLE III: Complexity analysis of BioTrust-QoS components

Component	Time Complexity	Space Complexity (per node)
MPTO clustering	$O(K \cdot N \cdot I)$	$O(N)$
Trust computation	$O(\deg(i))$	$O(\deg(i))$
ESSO CH election	$O(P \cdot D \cdot G)$	$O(P)$
M-TCFPA	$O(P \cdot G \cdot F)$	$O(P \cdot K)$
Route discovery (on-demand)	$O(L \cdot \Delta)$	$O(\text{number of paths})$
ML-DRNN prediction	$O(H \cdot L^2)$	$O(H \cdot L)$
IDT (Firefly-WOA)	$O(P \cdot G \cdot M)$	$O(P)$

IV. Simulation Setup, Performance Evaluation, and Comparative Analysis

A. Simulation Environment and Parameters

We implemented the full BioTrust-QoS protocol in NS-3.35 to evaluate its practical behaviour. We chose NS-3 after testing others because it is accurate, especially for physical and mobility modelling, and it met our needs. Our code and configurations are open source for those who want to try or verify our results.

To really stress-test things, we set up a tough simulation. To stress-test, we ran a rigorous simulation. Nodes are placed randomly within a 1500x1500-meter area, large enough for multi-hop routing but not so big the network breaks. We used the Random Waypoint mobility model, which captures varied movement. Node speeds range from 5 to 25 meters per second, covering walking to car-like speeds. This captures both slow- and fast-changing networks—some selfish (dropping packets to save battery) and some "black holes" (lying about routes and then dropping everything). We kept the attacker count between 2% and 10% (for a 100-node network, that's up to 10 bad actors). If we went higher, any protocol would collapse, so 10% is our ceiling.

Sixty UDP flows carried 512-byte packets—just the sort you'd find with a bunch of sensors or lightweight control messages. We set the network data rate at 2 Mbps, which matches what you'd see on 802.11b/g gear running in ad hoc mode. All the simulation details are summed up in Table III.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Each scenario was run 30 times with different random seeds to ensure statistical reliability. Confidence intervals are at 95%. Simulations ran for 2000 seconds, allowing nodes to roam fully and trust scores to stabilise.

TABLE IV: Simulation Parameters

Parameter	Value
Network area	1500 × 1500 m^2
Number of nodes	50, 100, 150, 200, 250
Mobility model	Random Waypoint (5–25 m/s)
Simulation time	2000 seconds
Transmission range	250 meters
MAC protocol	IEEE 802.11b (DCF) Data
rate	2 Mbps
Traffic type	60 CBR flows over UDP
Packet size	512 bytes
Attack types	Selfish, Black hole (2 to 10 attackers) Trust
update interval	10 seconds (low/medium), 5 seconds (high)

We ran each configuration 30 times with different random seeds to obtain statistically meaningful results. The confidence intervals we report are at the 95% level. A simulation run lasted 2000 seconds of simulated time, which gave nodes enough time to move across the entire area multiple times and allowed the trust values to converge.

B. Performance Metrics

We chose five key metrics, each aimed at something network ops people actually worry about.

1) Packet Delivery Ratio (PDR): The fraction of sent packets reaching their destinations, averaged across all 60 flows. High PDR indicates routing handled movement and attacks well.

2) Throughput: Measures how much data, in kilobits per second, is delivered on time. Late arrivals do not count.

3) Average End-to-End Delay: The average time a packet takes from sender to receiver, including hops, waits, processing, and retransmissions. This is crucial for live data like audio or video.

4) Packet Drop Ratio: The inverse of PDR, tracking drops separately helps identify causes such as congestion, attacks, or route disruptions. Our trust module attempts to determine the cause, but the raw number remains informative.

5) Attack Detection Ratio: This measures how effectively the IDS detects malicious actors. It's the number of correAttack Detection Ratio: Measures how effectively the IDS detects malicious nodes, calculated as correctly flagged malicious nodes over total malicious nodes. A perfect score is desired. values mean we're not wrongly accusing honest nodes. These two ratios together show the trust model's accuracy.

$$\text{Attack Detection Ratio} = \frac{\text{Number of correctly identified malicious nodes}}{\text{Total number of actual malicious nodes}}$$

A perfect score is 1.0. False negatives (missed attacks) are dangerous, so we prioritized this metric highly.

6) Benevolent Detection Ratio: Benevolent detection ratio is the ability of the system to correctly identify cooperative (trustworthy) nodes. It is defined as:

$$\text{Benevolent Detection Ratio} = \frac{\text{Number of correctly trusted benevolent nodes}}{\text{Total number of benevolent nodes}}$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

A high value means that the protocol does not falsely accuse honest nodes of being malicious. Both detection ratios together give a complete picture of the trust system's accuracy.

C. Benchmark Protocols

To see where BioTrust-QoS stands, we put it up against three big-name routing protocols representing different design philosophies.

1) Standard AODV: Standard AODV: The Ad hoc On-Demand Distance Vector (AODV) protocol is the default routing protocol for MANETs. It computes routes only when needed and uses sequence numbers to prevent loops. However, AODV ignores trust and quality of service (QoS) needs. Comparing to AODV sets the baseline for a bare-bones solution.

2) Evolutionary Self-Cooperative Trust (ESCT): We set all All protocols used default parameters from their original papers. Simulations ran in the same environment with identical traffic and mobility patterns to ensure fair comparisons.formance: Packet Delivery Ratio and Delay

Looking at the packet delivery ratio (PDR), our BioTrust-QoS protocol consistently outperforms all three baselines across all node densities we tested.

At 50 nodes, BioTrust-QoS achieved a PDR of 94.3%. In contrast, AODV reached 71.2%, ESCT 84.1%, and ETRS 86.2%. This is a 32.4% boost over AODV and about 9.3% over ETRS. Even in small networks, combining clustering and trust clearly helps.

3) Efficient Trust-Based Routing Scheme (ETRS): ETRS blends trust evaluation with clustering. It weighs trust and hop count to select routes but lacks machine learning-based node movement prediction. Comparing to ETRS highlights the benefits of our ML-DRNN prediction.

We configured all benchmark protocols with their default parameters as specified in the original papers. For fair comparison, we ran them in the same simulation environment with identical traffic and mobility patterns.

D. QoS Performance: Packet Delivery Ratio and Delay

The results for packet delivery ratio appear in Figure 2 (not shown here) and are summarized in Table IV. The proposed BioTrust-QoS protocol consistently outperformed all three benchmarks across every node density we tested.

At 50 nodes, BioTrust-QoS achieved a PDR of 94.3%. In contrast, AODV reached 71.2%, ESCT 84.1%, and ETRS 86.2%. This is a 32.4% boost over AODV and about 9.3% over ETRS. Even in small networks, combining clustering and trust clearly helps.

Now, crank that up to 250 nodes, and things get dramatic. BioTrust-QoS still delivered 86.7% of packets. But AODV plummeted to just 61.2%, ESCT fell to 73.4%, and ETRS only managed 75.2%. So now, the improvement over AODV is more than 41%, and we're 15% ahead of ETRS. Why? Because hierarchical clustering (MPTO) keeps most route-finding traffic local to each cluster, BioTrust-QoS scales much better.

TABLE V: Packet Delivery Ratio (%) for different network sizes

Nodes	AODV	ESCT	ETRS	BioTrust-QoS
50	71.2 ± 2.1	84.1 ± 1.8	86.2 ± 1.5	94.3 ± 1.2
100	68.4 ± 2.3	81.3 ± 1.9	83.7 ± 1.7	92.1 ± 1.3
150	65.1 ± 2.5	78.2 ± 2.1	80.3 ± 1.9	89.8 ± 1.4
200	63.3 ± 2.7	75.8 ± 2.2	77.9 ± 2.0	88.2 ± 1.5
250	61.2 ± 2.9	73.4 ± 2.4	75.2 ± 2.2	86.7 ± 1.6

So what's happening to AODV as the networAs the network grows, AODV floods the entire network with every route search. Adding nodes and increasing mobility causes control traffic to overwhelm the network. BioTrust-QoS avoids this by operating mostly within clusters.ure, but the gaps get even bigger. At 50 nodes, BioTrust-QoS had an average delay of 78 milliseconds. AODV lagged way behind at 295 ms—almost four times higher. ESCT and ETRS landed at 152 ms and 134 ms. So yes, compared to AODV, our delay dropped 73.5%.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

At 250 nodes, BioTrust-QoS's delay rose slightly to 95 milliseconds, while AODV's soared to 395 ms—a 75.9% reduction favoring BioTrust-QoS. AODV repeatedly tries sending data along broken routes, causing delays. Our ML-DRNN predictor identifies weak links early and starts rerouting about 1.5 seconds before a break. This proactive approach avoids scrambling to fix routes after failures.

E. Throughput and Packet Drop Ratio

Throughput measures how much data actually gets delivered to applications. Table V presents the results.

TABLE VI: Throughput (kbps) for different network sizes

Nodes	AODV	ESCT	ETRS	BioTrust-QoS
50	412 ± 15	478 ± 12	491 ± 11	563 ± 10
100	398 ± 18	465 ± 14	480 ± 13	548 ± 12
150	375 ± 20	451 ± 16	467 ± 14	534 ± 13
200	361 ± 22	438 ± 18	455 ± 16	521 ± 14
250	348 ± 24	428 ± 19	443 ± 17	508 ± 15

At 50 nodes, BioTrust-QoS hit 563 kbps—1At 50 nodes, BioTrust-QoS achieved 563 kbps—12.4% better than AODV. At 250 nodes, the gap grew to 14.2% (508 kbps vs. 348 kbps). Improvements over ESCT and ETRS were smaller but notable: about 7% at 50 nodes and 9% at 250 nodes. This steady performance as networks grow shows our ESSO-based cluster-head selection effectively distributes traffic load.tio shows how often packets are dropped.

Packet drop ratio tells the other side of the story. We measured this as the percentage of packets that never reached their destination, whether due to congestion, attacks, or routing failures. Figure 3 in the full paper shows the trends, but the key numbers are as follows.

At 50 nodes, BioTrust-QoS dropped only 5.7% of packets. AODV dropped 28.8%, ESCT 15.9%, and ETRS 13.8%. This is an 80% reduction compared to AODV. At 250 nodes, BioTrust-QoS lost 13.3%, while AODV rose to 38.8%. Even compared to ETRS, BioTrust-QoS drops fewer packets, nearly halving its loss rate at high node counts and higher densities. In short, more nodes mean more chances for packet collisions, even with good routing. BioTrust-QoS handles this better by constantly updating which nodes forward packets based on real-time link quality, but if you cram enough nodes into a mobile network, some losses are just inevitable. BioTrust-QoS handles this better than the others because the opportunistic forwarder selection adapts to real-time link quality, but no protocol can completely eliminate collisions in a dense, mobile network.

F. Security Performance: Attack and Benevolent Detection

We ran security tests by injecting both selfish and black-hole attackers into the network. The hardest scenario used 250 nodes, and we varied the number of attackers from two to ten. Here's the upshot:

TABLE VII: Attack Detection Ratio (%) at 250 nodes with varying attackers

Attackers	AODV	ESCT	ETRS	BioTrust-QoS
2	34.2 ± 5.2	78.4 ± 3.1	82.1 ± 2.8	91.3 ± 2.1
4	31.7 ± 5.5	75.9 ± 3.4	79.8 ± 3.0	89.7 ± 2.3
6	29.3 ± 5.8	73.2 ± 3.7	77.4 ± 3.2	88.2 ± 2.5
8	27.1 ± 6.1	70.8 ± 4.0	75.1 ± 3.5	86.9 ± 2.7
10	25.4 ± 6.4	68.5 ± 4.3	73.2 ± 3.8	85.4 ± 2.9

AODV doesn't perform any attack detection itself—the numbers we report there come from running our IDS alongside it. Its detection is weak because it never gathers trust information. BioTrust-QoS, in contrast, caught more than 90% of attackers even

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

with just two bad nodes. Even with 10 (about 4% of the network), it still caught 85.4%. This is 60 percentage points better than AODV at the toughest setting, and nearly 17% better than ESCT.

The benevolent detection ratio, shown in Table VII, measures how well the protocol avoids false accusations. This is just as important as detecting real attacks, because falsely accusing honest nodes can paralyze the network.

TABLE VIII: Benevolent Detection Ratio (%) at 250 nodes

Attackers	AODV	ESCT	ETRS	BioTrust-QoS
2	72.5 ± 3.1	85.3 ± 2.2	88.1 ± 1.9	95.8 ± 1.4
4	71.8 ± 3.3	84.7 ± 2.4	87.4 ± 2.0	95.2 ± 1.5
6	71.2 ± 3.5	84.1 ± 2.6	86.9 ± 2.1	94.7 ± 1.6
8	70.5 ± 3.7	83.5 ± 2.8	86.3 ± 2.2	94.1 ± 1.7
10	69.9 ± 3.9	82.9 ± 3.0	85.7 ± 2.3	93.5 ± 1.8

Benevolent detection ratio is equally important to avoid falsely blaming honest nodes. BioTrust-QoS accurately recognizes honest nodes at least 95.8% of the time, even with attackers present. With 10 attackers, it remains above 93.5%. This is over 10% better than AODV. With a false-positive rate of 6.5%, honest nodes are rarely blacklisted multi-level trust model. In the high-risk setting we used, our protocol checks both direct behavior and recommendations from several neighbors. One bad actor can't frame someone; it takes a majority to make an accusation stick. Plus, the Beta reputation model smooths out one-off issues, so a single lost packet won't ruin a node's standing.

What makes the difference? We traced this back to the multi-level trust model. In the high-level mode (which we used for these experiments because attackers were present), the protocol combines direct observations with weighted recommendations from multiple neighbors. A single malicious node cannot falsely accuse an honest node because the protocol requires consensus from several recommenders. The IDT model with Beta reputation also smooths out short-term fluctuations, so a temporary packet loss does not permanently ruin a node's reputation.

V. CONCLUSION

Our simulations revealed key takeaways. First, combining hierarchical clustering with bio-inspired cluster-head selection lets BioTrust-QoS scale to 250 nodes without failure. Second, the ML-DRNN predictor enables route fixes before breaks, reducing end-to-end delay by nearly 75% compared to AODV. The trust assessment detects over 85% of attackers—even in crowded, mobile environments—while keeping false accusations under 7%. Finally, opportunistic forwarding maintains high throughput, exceeding 500 kbps in large scenarios despite heavy attacks.

REFERENCES

[1] R. Kocherla, M. Chandra Sekhar, and R. Vatambeti, "Enhancing the energy efficiency for prolonging the network life time in multi-conditional multi-sensor based wireless sensor network," Journal of Control and Decision, vol. 10, no. 1, pp. 72–81, 2023. doi: 10.1080/23307706.2022.2057362

[2] S. Venkatasubramanian, A. Suhasini, and C. Vennila, "Cluster head selection and optimal multipath detection using coral reef optimization in MANET environment," International Journal of Computer Network and Information Security, vol. 14, no. 3, pp. 88–99, 2022. doi: 10.5815/ijcnis.2022.03.07

[3] R. Suresh Kumar, P. Manimegalai, V. Raj, R. Dhanagopal, and A. Johnson Santhosh, "Cluster head selection and energy efficient multicast routing protocol-based optimal route selection for Mobile Ad Hoc Networks," Wireless Communications and Mobile Computing, vol. 2022, art. no. 5318136, 2022. doi: 10.1155/2022/5318136

[4] N. Boddu, V. Boba, and R. Vatambeti, "A novel georouting potency based optimum spider monkey approach for avoiding congestion in energy efficient mobile Ad-Hoc network," Wireless Personal Communications, vol. 127, pp. 1157–1186, 2022. doi: 10.1007/s11277-021-08571-4

[5] M. Vigenesh and R. Santhosh, "An efficient stream region sink position analysis model for routing attack detection in Mobile Ad Hoc Networks," Computers & Electrical Engineering, vol. 74, pp. 273–280, 2019. doi: 10.1016/j.compeleceng.2019.02.005

[6] M. A. Tawfeeq, "Optimizing cluster head selection in Mobile Ad Hoc Networks: A connectivity probability approach using Poisson distribution and residual energy," Inge´nierie des Syste`mes d'Information, vol. 28, no. 5, pp. 1353–1359, 2023. doi: 10.18280/isi.280524

[7] K. Sindhanaiselvan, J. M. Mannan, and S. K. Aruna, "Designing a dynamic topology (DHT) for cluster head selection in mobile adhoc network," Mobile Networks and Applications, vol. 25, pp. 576–584, 2020. doi: 10.1007/s11036-019-01283-x

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

[8] M. A. Jubair, S. A. Mostafa, D. A. Zebari, H. M. Hariz, N. F. Abdulsattar, M. H. Hassan, A. H. Abbas, F. H. Abbas, A. Alasiry, and M. T. H. Alouane, "A QoS aware cluster head selection and hybrid cryptography routing protocol for enhancing efficiency and security of VANETs," IEEE Access, vol. 10, pp. 124792–124804, 2022. doi: 10.1109/ACCESS.2022.3224466

[9] R. Vatambeti, S. Sanshi, and D. P. Krishna, "Optimized Routing by Combining Grey Wolf and Dragonfly Optimization for Energy Efficiency in Wireless Sensor Networks," Applied Sciences, vol. 12, no. 21, art. no. 10948, 2022. doi: 10.3390/app122110948

[10] H. A. Younus and C. Koc¸ak, "Optimized Routing by Combining Grey Wolf and Dragonfly Optimization for Energy Efficiency in Wireless Sensor Networks," Applied Sciences, vol. 12, no. 21, 10948, 2022. doi: 10.3390/app122110948

[11] U. Srilakshmi, S. A. Alghamdi, V. A. Vuyyuru, N. Veeraiah, and Y. Alotaibi, "A Secure Optimization Routing Algorithm for Mobile Ad Hoc Networks," IEEE Access, vol. 10, pp. 14260–14269, 2022. doi: 10.1109/ACCESS.2022.3144679

[12] V. Sundaram and G. Ravi, "Optimized Geographic Routing in Mobile Ad Hoc Network Using Gray Wolf Optimization," EAI Endorsed Transactions on Energy Web, 2020. doi: 10.4108/eai.13-7-2018.162832

[13] S. Saravanan, D. Prabakar, and S. S. Sathya, "Trust aware ad hoc routing protocol with key management based mechanism and optimal energy-efficient cluster head selection in mobile ad hoc networks," Concurrency and Computation: Practice and Experience, 2023. doi: 10.1002/cpe.7599

[14] C. Edwin Singh, S. Sharon Priya, B. Muthu Kumar, K. Saravanan, A. Neelima, and B. Gireesha, "Trust aware fuzzy clustering based reliable routing in Manet," Measurement: Sensors, vol. 29, art. no. 100869, 2023.

[15] P. Bondada, D. Samanta, M. Kaur, and H. N. Lee, "Data security-based routing in MANETs using key management mechanism," Applied Sciences, vol. 12, no. 3, p. 1041, 2022. doi: 10.3390/app12031041

[16] L. V. Rajathi, "An advancement in energy efficient clustering algorithm using cluster coordinator-based CH election mechanism (CCCH)," Measurement: Sensors, vol. 25, art. no. 100623, 2023. doi: 10.1016/j.measen.2022.100623

[17] T. Hai, J. Zhou, Y. Lu, D. Jawawi, D. Wang, E. M. Onyema, and C. Biamba, "Enhanced security using multiple paths routine scheme in cloud-MANETs," Journal of Cloud Computing, vol. 12, no. 1, p. 68, 2023. doi: 10.1186/s13677-023-00443-5

[18] M. M. Mukhedkar and U. Kolekar, "Trust-based secure routing in mobile ad hoc network using a hybrid optimization algorithm," The Computer Journal, vol. 62, no. 10, pp. 1528–1545, 2019. doi: 10.1093/comjnl/bxz061

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

[19] K. Kumaresan, C. Rohith Bhat, and K. Lalitha Devi, "A novel fuzzy marine white shark optimization based efficient routing and enhancing network lifetime in MANET," Wireless Personal Communications, 2023, pp. 1–23. doi: 10.1007/s11277-023-10675-y
[20] B. Sharma and D. Saxena, "Design and analysis of energy efficient service discovery routing protocol in MANETs," SN Computer Science, vol. 4, no. 5, p. 495, 2023. doi: 10.1007/s42979-023-01899-7
[21] G. R. Devi, M. S. Das, and M. R. Murthy, "Secure cross-layer routing protocol with authentication key management scheme for manets," Measurement: Sensors, vol. 29, art. no. 100869, 2023.
[22] S. B. Ninu, "An intrusion detection system using exponential henry gas solubility optimization based deep neuro fuzzy network in MANET," Engineering Applications of Artificial Intelligence, vol. 123, art. no. 105969, 2023. doi: 10.1016/j.engappai.2023.105969
[23] A. M. Desai and R. H. Jhaveri, "Secure routing in mobile Ad hoc networks: a predictive approach," International Journal of Information Technology, vol. 11, no. 2, pp. 345–356, 2019. doi: 10.1007/s41870-018-0188-y
[24] N. S. Farheen and A. Jain, "Improved routing in MANET with optimized multi-path routing fine-tuned with hybrid modeling," Journal of King Saud University - Computer and Information Sciences, 2020. doi: 10.1016/j.jksuci.2020.01.001
[25] R. Sahu, S. Sharma, and M. A. Rizvi, "ZBLE: zone-based efficient energy multipath protocol for routing in mobile Ad Hoc networks," Wireless Personal Communications, vol. 113, no. 4, pp. 2641–2659, 2020. doi: 10.1007/s11277-020-07345-8
[26] N. Veeraiah, O. I. Khalaf, C. V. P. R. Prasad, Y. Alotaibi, A. Alsufyani, S. A. Alghamdi, and N. Alsufyani, "Trust aware secure energy-efficient hybrid protocol for manet," IEEE Access, vol. 9, pp. 120996–121005, 2021. doi: 10.1109/ACCESS.2021.3108807
[27] B. Muneeswari and M. S. K. Manikandan, "Energy efficient clustering and secure routing using reinforcement learning for three-dimensional mobile ad hoc networks," IET Communications, vol. 13, no. 12, pp. 1828–1839, 2019. doi: 10.1049/iet-com.2018.6150
[28] B. A. Kumar, M. V. Subramanyam, and K. S. Prasad, "An energy-efficient clustering using k-means and AODV routing protocol in ad-hoc networks," International Journal of Intelligent Engineering and Systems, vol. 12, no. 2, pp. 117–127, 2019.
[29] K. Lavanya, R. Indira, A. K. Velmurugan, and M. Janani, "Mobility-based optimized multipath routing protocol on optimal link state rout- ing in MANET," in 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), IEEE, 2023, pp. 1–6. doi: 10.1109/ICAISC58445.2023.10199484
[30] S. Aluvala and K. Rajasekhar, "Secure routing in MANETS using adaptive cuckoo search and entropy-based signature authentication," Wireless Personal Communications, vol. 128, no. 3, pp. 1519–1541, 2023. doi: 10.1007/s11277-022-10008-5
[31] A. Shajin Nargunam and M. P. Sebastian, "Self-organized QoS aware multicast routing scheme for Ad hoc networks," International Journal of Computers and Applications, vol. 32, no. 1, 2010. doi: 10.2316/Journal.202.2010.1.202-2390
[32] A. Shajin Nargunam and M. P. Sebastian, "QoS-Aware Multicast Routing for Mobile Ad Hoc Networks," International Journal of Business Data Communications and Networking, vol. 4, no. 2, 2008. doi: 10.4018/jbdcn.2008040101
[33] F. Hamza and S. M. C. Vigila, "An energy-efficient cluster head selection in MANETs using emperor penguin optimization fuzzy genetic algorithm," in Proceedings of International Conference on Recent Trends in Computing: ICRTC 2022, Ghaziabad, India, pp. 453–468, 2023. doi: 10.1007/978-981-19-8825-7_39
[34] V. K. Mahadevachar and N. T. Hosur, "Metaheuristic based energy efficient routing protocol in MANET using battle royale optimization," International Journal of Intelligent Engineering and Systems, vol. 16, no. 4, 2023. doi: 10.22266/ijies2023.0831.01
[35] R. Vatambeti and G. Mamidisetti, "Routing attack detection using ensemble deep learning model for IIoT," Information Dynamics and Applications, vol. 2, no. 1, pp. 31–41, 2023. doi: 10.56578/ida020104
[36] A. A. Abdelhamid, S. K. Towfek, N. Khodadadi, A. A. Alhussan, D. S. Khafaga, M. M. Eid, and A. Ibrahim, "Waterwheel plant algorithm: A novel metaheuristic optimization method," Processes, 2023.