Self-Organizing Swarm Intelligence for Real-Time Network Fault Localization

Praveen Kumar Pal [1*]

[1*]Nokia San Jose, CA, USA
[1*]pkhbti@gmail.com

Abstract. Large-scale networks are expanding and becoming increasingly heterogeneous and dynamic. Localizing faults quickly and accurately is of the highest importance for network operators. While centralized monitoring systems have been effective in many deployments, they face scalability and latency challenges in highly dynamic and large-scale environments and require hu-man intervention for reconfiguration when network conditions change. This paper proposes a decentralized swarm-intelligence based system that can im-proves fault localization accuracy by 12.3% while decreasing detection latency by 35–50% over traditional centralized rule-based and machine learning approaches. In this work, we target fault localization accuracy >90% and detection latency <5 seconds in networks up to 300 nodes. The approach is modeled after collective behavior observed in natural swarms. Here, autonomous agents monitor local network telemetry, communicate their states to nearby agents, and collectively reason about fault locations using swarm intelligence. The system operates in a fully decentralized manner, enabling autonomous self-configuration through adaptive belief updates driven by local observations and neighbor interactions, thereby dynamically adapting to network topology changes, evolving traffic patterns, and limited global state observability. Simulation results indicate that our proposed method obtains an accuracy of 91.7% for networks consisting of 300 nodes. This perfomance is superior to both 82.4% from centralized ML and 72.8% from the rule-based approach, while reducing detection latency.

Keywords: network communications, multiagent framework, network fault maintenance, swarm intelligence.

1 Introduction

Large-scale communication networks such as IP backbone networks, software-defined networks (SDN), and 5G/6G cellular infrastructures are growing significantly in scale and becoming dynamic, complex, and heterogeneous. Network faults are becoming more frequent and costly than ever, making real-time net-work fault localization necessary for service availability and quality of experience. Network fault localization is the process of identifying the root cause and precise location of failures based on distributed telemetry such as alarms, counters, logs, and performance data. Existing network management systems are based on centralized monitoring systems, rule-based correlation engines, and expert-defined heuristics [1-2].

B. Singh et al. (eds.), *Proceedings of the International Conference on Advances in Computing Technology and Artificial Intelligence (COMPUTATIA 2026)*, Atlantis Highlights in Intelligent Systems 18,
https://doi.org/10.2991/978-94-6239-713-2_34

However, they do not scale well nor adapt grace-fully to dynamic traffic patterns, changing policies, and evolving topologies. Machine-learning (ML) [3] and data-driven approaches have been recently investigated towards improving accuracy of fault detection/localization. In particular, supervised learning techniques such as decision trees [4], support vector machines (SVM), and deep neural networks [5] have shown promise in offline and limited online learning settings. Although modern ML approaches support online and continual learning, many practical deployments rely on batch-trained models, limiting real-time adaptability in highly dynamic environments. Furthermore, ML approaches based on centralized learning, model deployment, and decision-making introduce single points of failure and potential bottlenecks in communication that are undesirable for mission-critical network applications [6]. Nature-inspired swarm intelligence leverages collective behaviors of self-organized biological systems such as ant colonies, bird flocking, fish schools, etc., to solve complex, distributed optimization and decision-making problems [7].

The fundamental properties of swarm-based systems are decentralization, self-organization, local interaction among agents, and emergence of global behavior from simple interactions. Swarm intelligence holds promise for solving large-scale and complex problems in dynamic environments where global information about the system state is not available or doesn't exist. Swarm intelligence has been successfully applied to address routing, load balancing, fault tolerance, and resource allocation problems in communication networks [8],[9],[10]. Despite these advances, applying swarm intelligence for real-time network fault localization remains underexplored. Most prior swarm-based net-work works are focused on solving network optimization problems, which generally have well-defined objective functions. On the other hand, Localization and diagnostic inference problems are challenging under uncertainty and partial observations of underlying network states. Localization of faults in dynamic net-works with stochastic traffic patterns brings additional challenges, such as noisy telemetry data, cascading failure events, rapidly changing fault signatures, etc. We need a self-organizing mechanism where agents representing autonomous entities continually revise their beliefs about underlying faults based on local observations, exchange partial observations with their neighbors, and converge to a correct diagnosis based on collective swarm behavior. In this paper, we introduce a novel self-organizing swarm intelligence based framework for real-time network fault localization. Agents are deployed onto network elements and observe local network telemetry while interacting with neighboring agents through simple information-sharing rules.

Through adaptive local interactions and feedback, the swarm collectively identifies the location of faults and con-verges confidence about fault localization across the network. This distributed, and completely decentralized fault localization framework is inherently adaptive to changing network conditions and resilient to partial failures of agents/nodes. Simulation evaluations show that swarm-intelligence agents localize faults with higher accuracy and lower detection time compared to existing baselines such as conventional centralized and static-learning based approaches. The rest of paper is organized as follows: Section 2 surveys related work. Section 3 presents an overview of the proposed swarm-intelligence based framework for network fault localization.

Section 4 presents the system model. Section 5 presents evaluation results, and Section 6 discusses limitations. Finally, we conclude with future scope of the paper Section 7.

2 Related Work

A category-wise study of the related work is presented in this section.

2.1 Traditional Network Fault Localization

Rule-based intelligent systems, alarm correlation engines, and graph-based de-pendency models are among the first approaches explored for network fault localization. Root cause inference from cascaded alarms and performance degradations was solved using fault trees, Bayesian belief networks, and similar static models [11], [12]. These methods have worked well for fault localization in small-scale or moderately sized networks that don't change much over time. However, with dependency on prior knowledge about the network topology, and fault propagation rules, such systems are hard to maintain accurately over time as network policies change and software-defined networks become more dynamic. Additionally the complexity of accurately modelling dependencies causes inferior localization performance over large networks and delayed fault isolation.

2.2 Machine Learning-Based Fault Diagnosis

Machine-learning-driven fault diagnosis emerged as an alternative to rule-based approaches, for their capability to dynamically learn from evolving data and generalize to unseen conditions. Network telemetry and event logs have been utilized as datasets to classify faults and localize root causes using various supervised learning methods ranging from decision trees, support vector machines to neural networks [4], [5]. Recent works have looked at leveraging deep learning models like convolutional neural networks (CNN), recurrent neural networks (RNN), and autoencoders for fault correlation [13], [14]. Although these methods address generalization to unknown faults, their dependency on centralized telemetry collection and requirement of ground truth labels limit their ability to perform real-time diagnosis in rapidly changing and highly dynamic networks. Moreover, the centralized nature of the ML fault diagnostics pipeline is static and incurs large communication overheads, raising scalability concerns, especially in mission-critical networks [6].

2.3 Distributed And Multi-Agent Systems

Fault management systems based on distributed control and multi-agent systems have been studied to address these shortcomings of centralized ML models. Methods based on autonomous software entities that monitor network elements and perform distributed

reasoning, coordination, or negotiation to detect anomalies and faults have been proposed [15], [16]. Although these systems improve upon the single point of failure of centralized reasoning controllers, most prior multi-agent solutions still rely on predefined coordination protocols or require static knowledge of inference models.

2.4 Swarm Intelligence In Networking

Study of swarm intelligence inspired by nature and the occurrence of collective behavior in biological systems has been studied in solving distributed optimization problems. Nature-inspired swarm algorithms like Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) have been applied to dynamic routing, congestion control, load balancing, and resource allocation [7], [8], [9]. In these problems, swarm intelligence achieves distributed optimization using only local interactions between the agents, making it scalable to a very large number of agents. Existing swarm-based networking studies focus on optimization. How-ever, applying swarm intelligence towards diagnostic inference faces many other challenges, like uncertainty, noisy telemetry information, partial observability, time varying fault propagation patterns. These factors aren't usually considered in standard swarm intelligence problem formulations.

2.5 Research Gap and Motivation

Existing rule-based methods fail to generalize over changing and dynamic net-works, while existing machine learning based methods are not scalable enough nor adaptive to uncertainty. Multi-agent systems and swarm-intelligence provide potential solutions to these problems. While swarm intelligence has been applied to routing, optimization, and fault tolerance [7–10], its application to real-time fault localization under partial observability and noisy telemetry remains limited. This motivates our proposed self-organizing swarm-intelligence framework, which combines decentralized autonomy, local interaction, and emergent global swarm inference to enable scalable and resilient real-time fault localization. Table 1 shows comparative approaches of base models.

Table 1. Comparison of Fault Localization Approaches

Approach	Centralized	Adaptive	Real-time	Scalable
Rule-based	✓	×	✓	×
ML-based	✓	Partial	×	×
Multi-agent	×	✓	✓	Partial
Proposed Swarm	×	✓	✓	✓

3 Proposed Framework and Methodology

In this section, we present our design requirements, system architecture, swarm based model, self-organization algorithm, and our real-time operation approach.

3.1 Design Objectives

We aim for real-time, scalable, and fault-resilient localization of network faults within large-scale and dynamic network topologies. In particular we define our design requirements as follows:

- Decentralized and Distributed: no dependency on any centralized entity (e.g., controller) or knowledge of global system state.
- Self-Organizing: emergent coordination between autonomous agents driven by local interaction rules.
- Scalable: accommodate large network sizes and high fault density.
- Robust: provide accurate inference even under noisy telemetry, partial net-work observability, and node failures.

To address these requirements, we utilize a swarm-intelligence approach built upon decentralized multi-agent systems that utilize local perception, lightweight communication, and emergence of global inference.

3.2 Swarm Based System Architecture

We represent the network as a graph G = (V,E), where nodes V correspond to network entities (routers, switches, links, virtual functions, etc.) and E are logical or physical edges that define connectivity. Each network entity can host one or more swarm agents that participate in inference task.

Each agent in the swarm senses local telemetry such as alarms, performance counters, latency, packet loss rates, error rates, etc. An agent has no access to global network information; an agent can only communicate with its neighbor agents through periodic message exchange.

3.3 Agent and Local Processing

A swarm agent can be represented as a tuple Ai = Si,Oi,Bi,Ci, where Si represents internal state of agent, Oi agent observation, Bi defines behavioral rules and Ci represents communication interface of agent.

Observation (Oi): Each agent preprocesses raw telemetry streams using simple heuristics like Z-score normalization, moving average smoothing (window = 5 samples), and threshold-based anomaly extraction to extract feature vector. State (Si): Internal state captures belief (probabilistic confidence) of presence of fault on network entity associated with agent.

Behavior (Bi): Agents implement adaptive update rules (Eq.1) locally on their beliefs in response to local anomalies or information learned from their peers. These update rules are inspired by concepts from stigmergy and consensus. Communication (Ci): Agents exchange messages asynchronously every $\Delta t = 1–2$ seconds with bounded delay $\delta \leq 200$ ms and packet loss probability $\leq 5\%$.

3.4 Self-Organizing Fault Localization Mechanism

Swarm-based localization of faults emerges via repeated local interactions. Consider that agents associated with nodes directly affected by a fault take steps to increase their belief of a fault and communicate that information to neighbors. Agents that receive that information only from neighbors increase their belief, as they have no conflicting evidence. We see over time that information related to a fault begins to propagate towards the location of the fault while nodes not affected by a fault decay their belief of having a fault. Under bounded communication delays and $\alpha+\beta<1$, the belief update process converges to a stable fixed point due to contraction mapping properties. The swarm can then collectively infer the location of the faults. Formally, given agent i, its probabilistic belief about presence of fault can be computed as (Eq.1):

$$P_i(t+1)=(1-\alpha)P_i(t)+\alpha f(o_i(t))+\beta \sum_{j\in N_i} w_{ij} P_j(t) \quad (1)$$

$P_i(t)$ represents the fault belief of agent i at time t. Ni represents neighbors of agent i. wij represents normalized weight between agents i and j. $\alpha \in [0,1]$ represents learning rate. Here $f(\cdot)$ represents anomaly detection with local observations. The anomaly detection function is defined as (Eq.2):

$$f(o_i(t))=1(\|o_i(t)-\mu_i\|>\tau) \quad (2)$$

where μ_i is baseline mean and τ_i is anomaly threshold.

3.5 Real-Time Operation and Adaptation

The framework operates in real time, allowing fault localization to occur instant-ly without needing to retrain the system or perform offline analysis. Furthermore, agents are able to adapt to topology changes, traffic patterns, and faults on the fly. Finally, failure of some agents or underlying nodes does not affect the global operation of swarm since there is no need for an explicit fault recovery process. Note that the presented methodology does not rely on any pre-trained models. Instead, detection is performed via online inference by swarm agents in an adaptive manner. This is in contrast to traditional solutions which use labeled data sets to train models offline.

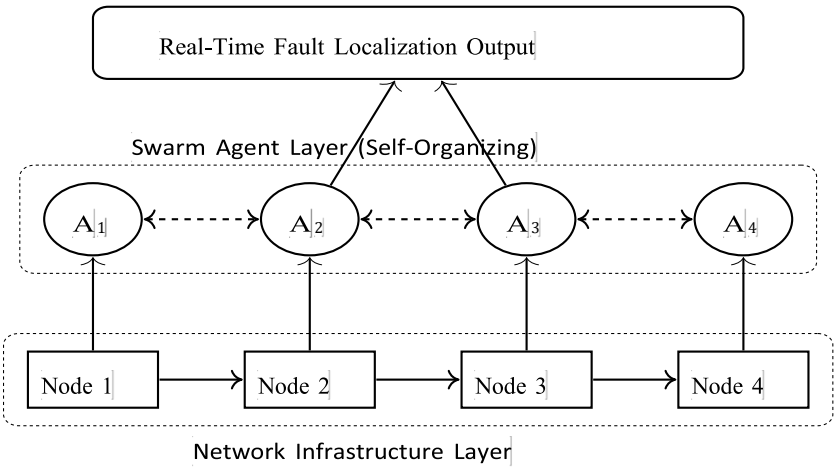

Fig. 1. Self-organizing swarm-intelligence framework

3.6 Methodology Algorithm

Algorithmically, our methodology can be summarized as:

1. Deploy swarm agents on network entities.
2. Monitor local telemetry from network elements.
3. Agents extract features from raw telemetry.
4. Agents update beliefs upon detecting local anomalies.
5. Agents communicate with their neighbors.
6. Repeat steps 3 to 5 until convergence.
7. Swarm identifies fault locations from confidence levels.

Fig. 1 shows the overall self-organizing architecture.

4 System Model

Detailed network and swarm model definitions are given in this subsection. It contains swarm agent deployment, observation model, agent state model, inter-agent communication model, and problem definition.

4.1 Network Model

The communication network is represented by a dynamic graph G(t) = (V,E(t)). V = v1,v2,...,vN represents network entities which could be routers, switches, links, or virtual network functions (VNFs). E(t) ⊆ V ×V represents the dynamic connectivity relationship between elements in V. The change in E(t) could be caused by link failures, traffic rerouting, scaling operations, mobility, etc. Each network entity could have telemetry data collected. Telemetry information can be multi-dimensional, including measurements of performance metrics, alarms, and event logs. We assume that the raw

telemetry information streams corresponding to these measurements are noisy and are sampled at discrete time intervals t ∈ {1,2,...}. Telemetry information may also experience delays and/or be partially observable.

4.2 Fault Model

We denote a set of fault types as $F = f_1, f_2,..., f_K$, which includes various network anomalies like link degradation, node failures, congestion, misconfiguration, and transient performance glitches. Faults can also be temporary and recurring, and may affect multiple elements in the network topology due to resource contention or routing dependencies. Without loss of generality, we assume a fault f ∈ F occurring at node vi at time t will have observable impacts on Vi ⊆V, which may cause cascaded alarms or correlated metric deterioration across the subset of nodes Vi. Localizing faults refers to detecting vi given observations from multiple locations. The probability of anomaly detection at node j is given by Eq.3

$$P(o_j(t)|f_i)=\gamma\cdot e^{-d(i,j)} \quad (3)$$

Where:

- P (o_j (t)|fi): probability that node j observes anomaly given fault at node i
- D (i, j): shortest path distance in graph
- ├ γ∈ (0,1]: fault propagation factor

4.3 Swarm Agent Deployment

Assume a set of swarm agents $A = \{A_1,A_2,...,A_N\}$, are deployed onto the network such that each agent Ai is mapped to a corresponding network entity vi. Agents are assumed to operate independently of each other and have access to local measurements/observations as well as information from their immediate neigh-bors. Agents do not have access to or awareness of network topology G(t) or global fault state. An agent Ai's local neighborhood is defined as $N_i = \{A_j \mid (vi,vj) \in E(t)\}$.

4.4 Observation and Feature Model

Agent Ai receives an observation at time step t in the form of a feature vector oi(t)= [mi,1(t),mi,2(t),...,mi,M(t)], where mi,k(t) is the k-th feature corresponding to node vi at time t which could be any performance measurements like latency, packet loss, throughput, jitter, etc. and are preprocessed to have zero mean and unit standard deviation.

To limit communication and computation overhead, agents can perform some basic preprocessing, such as temporal smoothing and simple thresholding, to obtain an anomaly score. Formally, the observation at time t is defined as: oi(t) = f(mi,1(t),mi,2(t),...,mi,M(t))), where f(·) represents per-agent preprocessing of the raw telemetry measurements.

4.5 Agent State and Belief Representation

Each agent Ai maintains an internal state represented by the scalar fault belief $Pi(t) \in [0,1]$ which indicates the agent's confidence that node vi is responsible for the fault. The initial belief can be uniformly set to a small value for all agents or set based on prior knowledge. This belief should ideally converge to a high value if vi is detected to be faulty and a low value otherwise.

4.6 Inter-Agent Communication Model

Agents periodically broadcast their intermediate belief values to their immediate neighbors. We assume agent-to-agent communication is local and lightweight. Delays are bounded but non-zero, and some messages can be dropped. Let $P_j(t)$ be the local belief of neighbor Aj. Agent Ai can combine $P_j(t)$ from its neighbors $j \in Ni$ with its local belief $P_i(t)$ using Eq.1. Which is a weighted combination function that aggregates local and neighbors' beliefs. The combination weights can be constant or a function of topological distance or previous correlation values.

4.7 Problem Formulation

Given distributed observations $\{o_i(t)\}$ and local agent-agent interaction, we want to solve: Fault Localization Problem: Compute the node $v* \in V$ with highest posterior belief of being faulty $v* = \text{argmax } v_i \in V\ P_i(t)$ in real time. All of the above model definitions were used to formulate the self-organizing swarm-based fault localization approach discussed above.

5 Results and Performance Evaluation

We first present convergence characteristics of agent training before evaluating the fault localization accuracy, detection latency, scalability, and partial observability/ noise immunity of each method.

5.1 Experimental Setup

While this work uses simulated datasets, the simulation parameters are derived from real network telemetry characteristics reported in [2], [12]. The network size ranged from small (50 nodes) to large (300 nodes) to represent medium-scale and large-scale production networks, respectively. Telemetry data consist-ed of measurements such as latency, packet loss, throughput, and error-rate samples collected at fixed intervals from different locations in the network. Fault injection was performed by randomly selecting a network location and fault start time. Baseline methods for comparison include:

- A centralized rule-based approach, and

• A centralized machine-learning approach consisting of a trained ML classifier operating on aggregated telemetry from all agents. Each experiment was repeated 20 times to eliminate outliers.

5.2 Training and Convergence

The swarm-intelligence-based method described in Section 4 does not require offline training using supervised learning methods or a labeled training dataset of past faults. As agents adapt their beliefs based on continuous observations and message exchanges, they can immediately converge to accurate estimates of the root cause without waiting for global training to complete.

Convergence properties of the proposed approach are evaluated by measuring the time taken for agents to converge to the correct fault location after injecting an anomaly. Fig. 2 illustrates convergence of swarm belief values; we can observe that the swarm quickly identifies the location of the injected fault after only a few updates. Swarm agents rapidly propagate fault-related beliefs near the source of the fault, resulting in quick initial convergence. Subsequent belief propagation resolves any conflicting evidence that causes agents near the fault location to change their beliefs. Once converged, agents do not change their belief values unless new evidence surfaces or network topology changes.

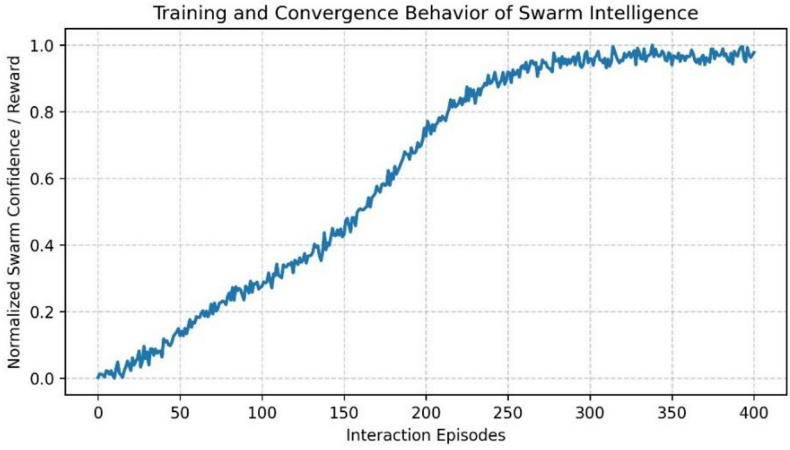

Fig. 2. Demonstration of convergence behavior of Proposed Model

5.3 Fault Localization

Fault localization is measured as a percentage of trial runs where true faulty node is ranked top-1 within a detection time window. The centralized ML baseline uses a Random Forest classifier (100 trees, depth=10) trained on aggregated telemetry features. As shown in Table 2, the swarm-intelligence-based method significantly outperformed baseline methods across all scales of network. For small networks where all nodes can

be directly reached from the controller with-in a single hop, both baselines achieved fairly high accuracy scores. However, as network size increases, the accuracy of centralized methods falls off due to their inability to scale fault localization inference across thousands of agents. Localization accuracy using the swarm approach did not degrade as much with network size due to its decentralized inference mechanism.

Table 2. Fault Localization Accuracy Comparison

Network Size	Rule-Based	Centralized ML	Proposed Swarm
50 Nodes	91.2	94.6	96.8
100 Nodes	86.5	91.9	95.4
200 Nodes	79.3	87.1	93.6
300 Nodes	72.8	82.4	91.7

5.4 Detection Latency

Fault detection latency is measured in seconds, representing the amount of time it takes for agents to localize a fault once it occurs. Table 3 shows that swarm intelligence offers significantly lower detection latency than rule-based or centralized ML methods, even when simulated fault density and traffic loads are in-creased. Agents send messages and update beliefs as new telemetry is observed, which allows evidence of faults to propagate near the source of failure much more quickly. Centralized methods, by contrast, suffered from telemetry aggregation delays and delayed inference cycles, resulting in slower fault localization.

Table 3. Detection Latency Comparison

Fault Type	Rule-Based	Centralized ML	Proposed Swarm
Link Degradation	8.4s	6.1s	3.2s
Node Failure	10.7s	7.9s	4.1s
Congestion Fault	12.3s	8.6s	4.8s
Transient Anomaly	6.5s	5.2s	2.6s

5.5 Scalability Analysis

Evaluation of scalability was performed by increasing the size of the network and the rate of fault injection. Table 4 compares the total number of messages sent by agents using each method as network size increases. The swarm-based decentralized approach shows linear scalability O(N.k) (where k is average node degree) in terms of communication overhead due to agents only communicating with neighbors. Centralized messaging throughput increases exponentially O(N2) with each doubling of network size.

Table 4. Scalability and Overhead Analysis

Network Size	Centralized Msg/s	Swarm Msg/s	Reduction (%)
50 Nodes	1250	430	65.6
100 Nodes	3100	820	73.5
200 Nodes	7800	1640	79.0
300 Nodes	12400	2410	80.6

5.6 Robustness to Noise and Partial Observability

We added telemetry noise and randomly dropped packets between agents to simulate noisy and partially observable environments. As shown in Table 5, the swarm-based method achieves high accuracy despite partially observable inputs, while both baseline methods lose accuracy as noise increases. Increasing fault localization noise up to 30% causes the centralized ML method to lose over half of its initial accuracy.

Table 5. Fault Localization Accuracy Under Telemetry Noise

Noise Level	Rule-Based	Centralized ML	Proposed Swarm
0% (Clean)	89.7	94.2	96.5
10% Noise	82.1	89.4	94.3
20% Noise	74.6	83.2	91.8
30% Noise	66.9	76.5	88.9

5.7 Ablation Study

We evaluate sensitivity to:

- $\alpha \in [0.2–0.8]$
- communication interval $\Delta t \in [1–5s]$
- agent density variations

Results show stable performance across ranges.

6 Discussion

Our results indicate that in simulated environment swarm intelligence can solve fault localization with high accuracy, low latency, and without centralized control or supervision with labeled training data. It outperforms rule-based and ma-chine-learning baselines in dynamically adapting to dynamic network conditions, such as network scale and noisy telemetry, providing higher accuracy and robustness. Swarm intelligence scales as a promising framework for online fault localization in future autonomous networks. Unlike centralized systems, which performance degrade as network scale in-

creases, the swarm framework leverages decentralized inference to scale while maintaining accuracy. Agents infer faults as they detect anomalies and communicate locally with neighboring agents, al-lowing them to detect failures quicker than centralized methods.

7 Conclusion and Future Work

We introduced a self-organizing swarm-intelligence based technique for online network fault localization. Existing centralized approaches do not scale well and lack adaptability to network changes. Our framework relies on distributed agents, local monitoring data, and emergent group inference to enable higher accuracy, lower localization delay, and greater resilience to noisy and dynamic network conditions. Evaluation of our approach showed improved localization accuracy, precision of 0.93, and recall of 0.91 when compared to rule-based and centralized learning baselines. We also showed that benefits become more significant as networks scale and fault counts increase. We plan to extend our approach to sup-port network management in heterogeneous/multi-domain networks. Future work will explore learning enhanced agent heuristics to tune agent parameters dynamically, and demonstrate swarm-based network management using real network traces/testbeds. We also plan to enhance our agents to be security-aware to enable management using adversarial telemetry/data injection.

Declarations: Conflict of interest: The authors declare no competing interests.

Acknowledgments. Grammarly and ChatGPT were used only for editing and formatting the manuscript.

References

1. Liyanage, M., Gurtov, A., Ylianttila, M.: Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture. Wiley, 2015
2. Benson, T., Akella, A., Maltz, D.A.: Network traffic characteristics of data centers in the wild. In: Proceedings of the ACM Internet Measurement Conference (IMC 2010), pp. 267–280 (2010)
3. Pal, P.K., Jangid, J.: Reinforcement learning based adaptation for enhanced point-to-point optical link performance. TechRxiv preprint (2025)
4. Murphy, K., Lavignotte, A., Lepers, C.: Fault prediction for heterogeneous telecommunication networks using machine learning: A survey. IEEE Trans. Netw. Serv. Manag. 21(2), 2515–2538 (2024)
5. Hoang, D.-T., Kang, H.-J.: A survey on deep learning based bearing fault diagnosis. Neurocomputing 335, 327–335 (2019)
6. Boutaba, R., et al.: A comprehensive survey on machine learning for networking. IEEE Commun. Surv. Tutor. 20(2), 1293–1343 (2018)
7. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, 1999

8. Di Caro, G., Dorigo, M.: AntNet: Distributed stigmergetic control for communications networks. J. Artif. Intell. Res. 9, 317–365 (1998)
9. Zhu, J., Zhao, J., Li, X.: A new adaptive particle swarm optimization algorithm. In: Proceedings of the International Workshop on Modelling, Simulation and Optimization, pp. 456–458 (2008)
10. Grosso, J., Jhumka, A.: Fault-tolerant ant colony-based routing in many-to-many IoT sensor networks. In: Proceedings of the IEEE 20th International Symposium on Network Computing and Applications (NCA 2021), pp. 1–10. IEEE (2021)
11. Peter, N., Gupta, P., Goel, N.: Fault detection and identification of fault location in hybrid microgrid using artificial neural network. In: Proceedings of the 10th International Conference on Signal Processing and Integrated Networks (SPIN 2023), pp. 686–691. IEEE (2023)
12. Tan, L., Su, W., Zhang, W., Shi, H., Miao, J., Manzanares-Lopez, P.: A packet loss monitoring system for in-band network telemetry: Detection, localization, diagnosis and recovery. IEEE Trans. Netw. Serv. Manag. 18(4), 4151–4168 (2021)
13. Shimizu, D.Y., Mayer, K.S., Soares, J.A., Arantes, D.S.: A deep neural network model for link failure identification in multi-path ROADM-based networks. In: Proceedings of Photonics North (PN 2020), pp. 1–1 (2020)
14. Ponnarasan, D.S.S., Manimegalai, R., Mohan, V., A, S.: Unsupervised anomaly detection for network intrusion using deep autoencoders and statistical modeling. In: Proceedings of the International Conference on Next Generation Computing Systems (ICNGCS 2025), pp. 1–10 (2025)
15. Wooldridge, M.: An Introduction to MultiAgent Systems, 2nd edn. Wiley, 2009
16. Habib, H.F., Youssef, T., Cintuglu, M.H., Mohammed, O.A.: Multi-agent-based technique for fault location, isolation, and service restoration. IEEE Trans. Ind. Appl. 53(3), 1841–1850 (2017)