Volume 1, 2025, Pages 13-21 *Journal of Computer and Communication Networks*
| Regular Article | Open Access

Environmental and iNet Driven Evolution of Adaptive Agents in Self Organizing Networks

Prabu Ragavendiran
Department of Computer Science and Engineering, Kangeyam Institute of Technology, Kangeyam, Tamil Nadu, India.
sdp.cse@builderscollege.edu.in

Article Info
Journal of Computer and Communication Networks
https://www.ansispublications.com/journals/jccn/jccn.html

https://doi.org/10.64026/JCCN/2025002

Received 05 November 2024
Revised from 20 December 2024
Accepted 06 January 2025
Available online 02 February 2025
Published by Ansis Publications.

Corresponding author(s):
Prabu Ragavendiran, Department of Computer Science and Engineering, Kangeyam Institute of Technology, Kangeyam, Tamil Nadu, India.
Email: sdp.cse@builderscollege.edu.in

Abstract – Self-Organizing Networks (SONs) represent a network design, which enables adaptive agents improve network performance through autonomous actions for self-management while reacting to shifting environmental conditions. The iNet mechanism provides the conceptual framework in this research. This research analyzes agent flexibility in SONs through evaluations of the iNet framework that operates with environmental assessment capabilities. The analysis uses simulated models to simulate agent conduct both with and without evolution in order to gauge how shifting network parameters affect performance indicators consisting of throughput, reaction time and load balancing. Both the agents' ability to adapt their behavior policies and the environmental assessment system jointly result in optimized resource allocation and improved management of tasks and better demand responsiveness. Research results prove that incorporating adaptive evolutionary mechanisms produces superior performance in SONs especially when the networks contain heterogeneous elements. The study introduces breakthrough findings about how evolutionary approaches and awareness of the environment help improve distributed network agents' behavior.

Keywords – Adaptive Agents, Self-Organizing Networks, Inet Evolution, Environmental Evaluation, Resource Utilization, Agent Adaptability, Simulation, Load Balancing, Performance Optimization.

I. INTRODUCTION

Self-organization is a comprehensive phrase that encompasses any form of autonomous reconfiguration of a system and represents the pinnacle of the hierarchy of technical structures. Self-Organizing Networks (SONs) will thus persist in enhancing forthcoming 6G communication systems, which must provide essential attributes such as resilience, cost-effectiveness, security, dependability, performance, scalability, stability, and functionality. SON structure is divided into a collection of smaller functional units known as SON Functions (SONFs). Examples are Energy Saving Management (ESM) and Coverage and Capacity Optimization (CCO) routines [1]. SONFs can engage in either helpful or negative interactions. Consequently, cutting-edge self-coordination abilities are essential to guarantee conflict-free functioning. The majority of research on 5G fails to examine self-coordination concepts during the design phase of new communication structures that use either reactive or predictive strategies.

Modern wireless communication networks use SONs to improve efficiency and performance thanks to their main beneficial quality. Network administrators do not need to contact SON systems for their autonomous management of network functions which include coverage and capacity. Network performance optimization through SON technologies makes use of continuous network condition analysis to dynamically change control parameters. Through steady network condition monitoring SON technologies detect and solve problems in real-time thus they extend system operation durations along with elevated user contentment levels. Self-healing capabilities of SONs represent their most significant strength because they enable networks to execute repair procedures without human intervention for ensuring service excellence [2]. The network becomes more reliable due to prevention methods that minimize the necessity of human intervention for troubleshooting.

The combination of SONs in wireless communication networks drives better operational performance and system efficiency to fulfill user needs at lower operational expenses.

The need for SONs appears essential because wireless networks produce complex network requirements that require fundamental management solutions. SONs deploy automated setups and maintenance operations to boost network capabilities as well as deliver improved user satisfaction results. The current standard network management struggles from two main problems since it depends on full human staff while needing extended timescales for parameter optimization and fault control activities. SON deployment enables automatic procedures that reduces the need for human personnel. The ability of SON to make quick decisions based on optimization through advanced algorithms and machine learning is possible because of its data evaluation scale capability [3]. The continuous analysis performed by SON generates automatic network changes that identify issues which produce enhanced reliability and increased capacity. Moreover, SONs facilitate the efficient deployment and management of small cells by network operators, thereby addressing the growing need for high-speed data services.

This research examines the effect that agent evolution mechanisms particularly iNet evolution and environmental evaluation facilities have on SON agent adaptability together with performance metrics. This research works to measure performance ramifications of adaptive evolutionary methods along with environmental understanding capabilities on network agent metrics such as throughput response time load balancing and resource usage and resource efficiency when operating in dynamic networks. The remaining sections of our study have been organized as follows: Section II reviews related works on SONs, and wireless network to further comprehend iNet evolution and environmental adaptation mechanisms. Section III describes our data collection method, ABM methodology, evolutionary dynamics and game theory, as well as statistical analysis and sensitivity testing. Section IV and V provides a detailed discussion of (i) assessment of iNet evolution process, and (ii) evaluation of EE facility. Lastly, Section VI concludes our study and proves that the integration of agent evolution from iNet and EE establishes effective agents in SONs.

II. RELATED WORKS

According to research by Fourati et al. [4], there are three primary possibilities for the architecture of Self-Organizing Network functionalities in cellular networks. These are classified as centralized, distributed, and hybrid architectures. Various SON functions may be executed by distinct architectures inside the same network. The examination of self-organizing complex networks has garnered considerable interest in recent years, especially with Internet of Things (IoT), wireless sensor networks (WSNs), and distributed systems. Self-organizing nodes use local affairs to create dynamic network connections which produce resilient yet effective communication in dynamic operating conditions. The original study of self-organizing networks examined the operation of decentralized control systems accompanied by local interaction mechanisms. The authors Piersa, Piekniewski and Schreiber [5] developed scale-free networks using preferential attachment to construct fail-safe frameworks which preserve network connectivity after node failure. The research provided basic knowledge for scientist to understand network system evolution through time utilizing localized rules.

Song, Zhang and Dolan investigated in [6] how decentralized local choice establishes global network resilience through evolving self-organizing network dynamics. In mobile ad-hoc networks (MANETs) the complex nodes form dynamic network connections by following both proximity rules and transmission power standards. The research work by Mills in [7] examines self-organizational capabilities in wireless sensor networks (WSNs) to prove how optimized communication protocols strengthen network longevity and operational effectiveness. Their research delivered a complete evaluation of distributed agent network self-organization methods to study fundamental mechanisms required for sustainable network operation. Almost all nodes require adaptive control systems that operate properly while network conditions alter.

The study by Dhabliya et al. [8] presented an adaptive protocol for mobile wireless sensor networks which uses adjustable transmission power control that responds to local network connectivity and restrictions on energy capacity. Progress in ML and AI technologies have enhanced the operating range of nodes within SONs. The authors developed reinforcement learning methods to optimize wireless sensor networks by controlling power and time allocation which optimizes network throughput. Research outcomes from network simulation demonstrate that the proposed transmission techniques enable superior local network throughput performance compared to greedy and random and cautious policies. In [9], Casagrande, Sassano, and Astolfi presented a Hamiltonian-based approach for enhancing connectivity and resilience in self-organizing networks, wherein nodes independently modulate their transmission power to attain a stable state characterized by near-complete connection and minimal energy consumption. This foundational research underpins the current study by offering a framework for the integration of AI-driven adaptive mechanisms.

Mészáros, Varga, and Kirsche [10] delineate multiple extensions to the preceding research on iNet. The scholars do not examine the evolutionary mechanism of iNet. Consequently, agent designers were required to meticulously and manually set antibodies within their agents during the design phase. Contrarily, iNet evolutionary approach enables agents to unconventionally modify their antibody arrangements during runtime, eliminating the need for manual adjustments. Dressler and Carreras [11] presents first simulation findings of iNet evolutionary approach; nonetheless, it does not examine the languages in the self-regulatory approach and BEYONDwork mechanism within iNet EE. The architecture of Bio-Networking resembles BEYOND in its application of biological concepts and mechanisms, enabling network applications to originally adjust to dynamic ecological changes inside the system. Nevertheless, its adaption engine differs from iNet. Although iNet is modeled after immunological responses, it utilizes a straightforward weighted sum computation for behavioral selections. Despite having an evolutionary approach, which dynamically modifies weighted values of the sum computations, agent modelers must still physically establish an equation for every behavior and determine a threshold value for every equation. Conversely, iNet necessitates no physical configuration efforts from agent modelers.

Artificial immune structures have been suggested and implemented in diverse application areas, including pattern recognition and anomaly detection. Pump, Ahlers, and Koschel [12] concentrates on the development of identifiers for non-self/self-classification and enhances the negative selection mechanism of artificial immune structure. They emphasize the precision in the pairing of antigens with its corresponding antibody. In contrast to previous works, this research recommends an artificial immune structure to enhance the autonomous flexibility of network applications. This work is the inaugural application of an artificial immune system in this arena. Furthermore, certain research [13] employing artificial immune systems expands upon the notion of danger signals. Boudec and Sarafijanović [14] presents a mechanism for detecting misbehavior nodes as an antigen according to the pattern of events within the routing process of ad hoc networks. Danger signals help decrease the false positive incidences (i.e., system misclassifies a well-functioning node as a malfunctioning one) by revising the definitions of typical occurrence sequences (self).

Conversely, iNet self-regulation procedure enables an agent to address both false negatives and false positives (i.e., the architecture is unable to detect unidentified non-self-antigens). BEYONDwork offers verbal and visual languages to constitute iNet, specifically for setting environmental conditions, detectors, and behavioral regulations. The linguistic efforts in BEYONDwork align with the current study on DSLs (domain-specific languages). Languages are regarded as area-certain Languages (DSLs) that concentrate on accurately representing the concepts and methods pertinent to a certain issue area. Numerous domain-specific languages (DSLs) exist for modeling biological systems, including biochemical networks, to facilitate the simulation and comprehension of these systems (e.g., [15]). Nonetheless, the purpose of BEYONDwork languages diverges from theirs; BEYONDwork languages are designed to replicate immunological (biological) systems enabling the development of autonomous and adaptable network applications.

III. DATA AND METHODS

Data Collection

This research examines the effect that agent evolution mechanisms particularly iNet evolution and environmental evaluation facilities have on SON agent adaptability together with performance metrics. This research works to measure performance ramifications of adaptive evolutionary methods along with environmental understanding capabilities on network agent metrics such as throughput response time load balancing and resource usage and resource efficiency when operating in dynamic networks.

Methodology: ABM

This research relies on Agent-Based Modeling (ABM) [16] as its main method to replicate distinct computing agents portraying nodes that operate within a Self-Organizing Network environment. Internal agent behavior adapts to the way agents interact with their surrounding peers as well as with their environmental conditions. Agents modify bandwidth assignment together with routing techniques while adjusting energy utilization through information received from their environment. The simulation system logs interactions between agents throughout time to study their adjustments regarding changing situations. The simulation tool allows Users to model complex network topologies with agent interactions which helps in thorough analysis of SON evolutionary processes. Through a utility function U_i the performance of agent i can be measured simultaneously against its throughput p_i and its related energy consumption e_i. The functioning of the network requires the utility function to observe Eq. (1).

$$U_i = a_1 \cdot p_i + a_2 \cdot \frac{1}{e_i} - a_3 \cdot (p_i^2 + e_i^2) \tag{1}$$

The coefficients α_1, α_2, α_3 let decision makers adjust which factors between throughput, energy efficiency and their combined performance determine the network evaluation results.

Evolutionary Dynamics and Game Theory

Evolutionary game theory serves as the approach to simulate agent competition and cooperation behaviors while modeling evolutionary elements of SONs. Each agent achieves evaluation through success in its ability to improve network performance as well as energy efficiency alongside system stability. The payoff matrix A allows an extension for agents i and j by adding dimensions representing cooperation relations and competitive relationships alongside mutual benefits. Agent i achieves the total payoff P_{ij} through playing strategy s_i when confronting agent j who plays strategy s_j, computed using Eq. (2).

$$P_{ij} = \sum_{k=1}^{m} A_{ijk} \cdot x_j \cdot \left(1 - e^{-\beta_k (p_i - p_j)^2}\right) \tag{2}$$

The payoff P_{ij} shows the interaction agent i shares with agent j whereas A_{ijk} defines the corresponding value for this connection. Agent $j's$ strategy is evaluated through strategy proportions x_j but agents i and j use p_i and p_j as their individual defense strategies examples include throughput. The sensitivity of payoffs to strategy difference depends on the parameter β_k as well as the number of interaction dimensions such as cooperation or competition which is represented by m.

Statistical Analysis and Sensitivity Testing
Statistical tests serve as validation methods for the simulation run results. The analysis uses t-tests alongside ANOVA to perform tests that determine performance variations between agent configurations and environmental circumstances. The network's performance together with its evolutionary patterns become subject to sensitivity testing under modifications of key parameters such as learning rates and environmental noise levels. Evaluating the total network T_{total} throughput demands the summation of weighted individual agent throughputs T_i, using Eq. (3), which decreases due to distance and agent capacity factors.

$$T_{total} = \sum_{i=1}^{n} \left(\gamma_i \cdot T_i \cdot e^{-\lambda_i (d_i - d_0)^2}\right) \tag{3}$$

Network throughput T_{total} results from weighting agent individual throughputs T_i at point i using factors γ_i which define network capacities and significances. All agents respond to outside elements through the λ_i coefficient which determines the extent of external variables on their production capacities. The distance d_i or a similar factor, influences the agent's performance, with d_0 acting as a reference distance for normalization. The expression unites n network components through their weighted throughputs T_i using elements γ_i.

IV. RESULTS

The section presents different simulated outcomes for agent adaptability assessment. It assesses the influence of an iNet evolution procedure on agents' adaptability. Additionally, it illustrates how the environment evaluation (EE) facility enhances the adaptability of agents. **Fig 2** represents Mean performance index at various mutation rates (RG = 2)

Assessment of iNet evolution process
This section illustrates the influence of the iNet evolution process on agent adaptation by differentiating simulation outcomes with/without iNet evolution. **Fig 3** illustrates the manner in which agents adjust their population in response to the workload variations represented in **Fig** 1. Through development, iNet enables agents to develop and modify their genetic behavior strategies. Consequently, as they acquire energy by handling service requests, they effectively execute reproduction or replication activities to augment their populations. They also appropriately exhibit mortality behavior to reduce their population as effort diminishes. Conversely, in the absence of evolution, agents do not transform their arbitrarily constituted genes during a model. Consequently, they are unable to adjust their workforce to fluctuations in workload.

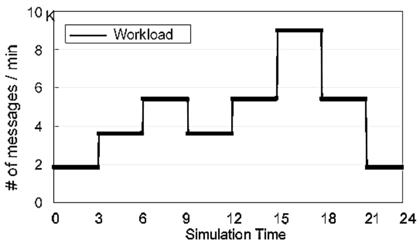

Fig 1. Workload

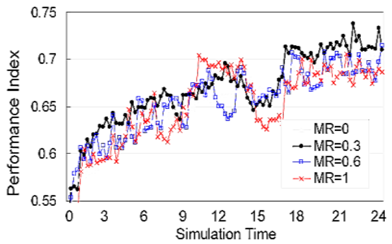

Fig 2. Mean Performance Index at Various Mutation Rates (RG = 2)

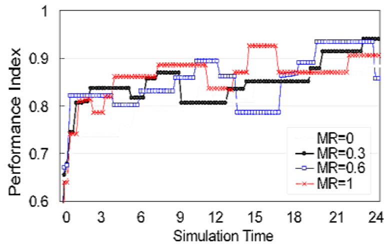

Fig 3. Optimal Performance Index with Varying Mutation Rates (RG = 2)

Fig 8 illustrates the manner in which agents adjust their throughput in response to variations in workload. Alterable agents unconventionally sustain higher throughput by vigorously modifying their population and locations through reproduction and migration behaviors. In the absence of evolution, agents are unable to change their throughput because to the lack of genetic evolution. **Fig 9** illustrates the manner in which agents diminish their feedback duration for the user. Initially, response time remains elevated as just 4 agents manage 2,000 requests every minute, compounded by the considerable distance between the user and the agents. Alterable agents gather sufficient energy from users and execute reproduction, replication, and migration actions, their response time diminishes significantly.

During workload surges at 3:00, response duration escalates to 16 seconds; still, agents mitigate it to roughly 4 seconds by adjusting their positions and personnel. Thereafter, they sustain a low response duration despite workloads surge at 6:00, 12:00, and 15:00. In the absence of evolution, agents are unable to sustain minimal response times. **Fig 10** illustrates the distance (mean quantity of hops) between users and agents. Firstly, 4 agents are arbitrarily positioned throughout the network, resulting in a considerable gap between them. Nevertheless, evolvable agents progressively diminish it by advancing towards the user. This modification of agent locations reduces user response time. In the absence of evolution, agents fail to adjust their positions during a simulation. **Fig 11** illustrates the distribution of workload among agents utilizing the Load Balancing Index (LBI).

The variation in the quantity of service requests assigned to an agent is denoted by LBI. A reduced LBI signifies a greater distribution of task across agents. Due to the uneven distribution of requests across agents, despite their attempts to duplicate for equitable processing, LBI escalates during peak workloads at 3:00, 6:00, 12:00, and 15:00. LBI increases as workload diminishes around 9:00, 18:00, and 21:00 due to some agents being idle and handling less requests. Evolvable agents promptly diminish LBI by altering their populace notwithstanding these augmentations. Agents are unable to generate low LBI during simulation without evolution. **Fig 12** depicts the distribution of resource use across hosts, using the Resource Use Balancing Index (RUBI) in Eq. (4).

$$\mathrm{RUBI} = \sqrt{\frac{\sum (R_k - \mu)^2}{N}} \tag{4}$$

N represents the quantity of active hosts, or the hosts on which agents are now functioning. The anticipated R (the ratio of resources employed by alleles to the total resources supplied by the active hosts) is denoted as R_k, reflecting the resource usage on $host_k$. The variability in resource use across active hosts is denoted by RUBI. A lower RUBI indicates more dispersion in resource consumption. Resource consumption escalates on the host inhabited by agents as their number expands in response to a heightened workload. Some, however, migrate to proximate hosts with enhanced resource availability. This signifies that agents want to distribute equally among hosts, which is why they reduce RUBI immediately after its increase. Agents cannot allocate resource use among hosts without evolution.

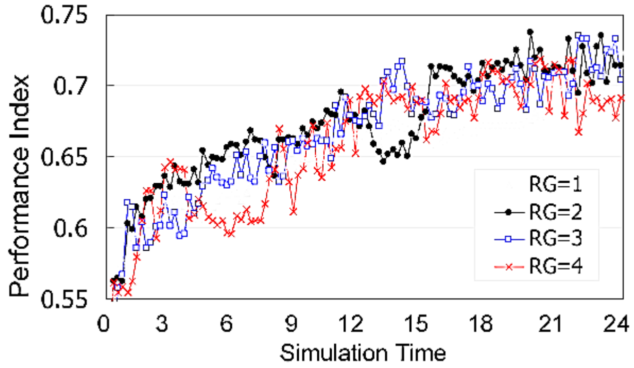

Fig 4. Average Performance Index (MR = 0.3) With Various Mutation Ranges

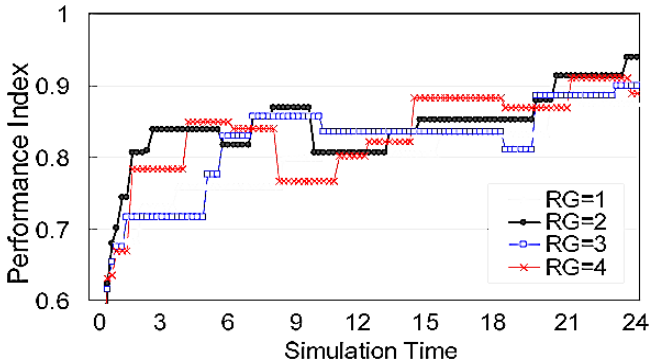

Fig 5. Maximum Performance Index (MR = 0.3) For Various Mutation Ranges

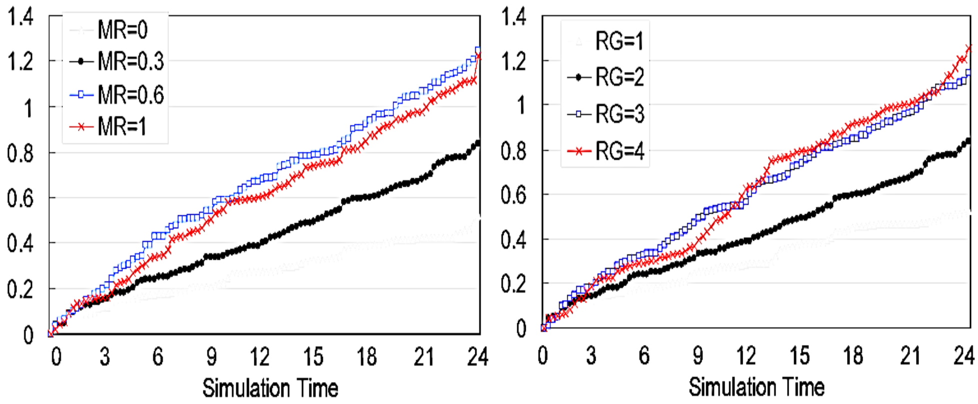

Fig 6. Total Difference Between the Consecutive Performance Index

Fig 4 illustrates the average agents' performance index. Agents consistently enhance their performance to 0.8 with evolution active, but do not exhibit improvement when evolution is deactivated. **Fig 5** illustrates that iNet facilitates agents in effectively growing and adapting to fluctuating network circumstances when considered with **Fig 7** to **12**. **Fig 14** illustrates the variation of the agents' performance index. A reduced variance shows that agents provide more consistent index findings. Agents achieve minimal variance while evolution is active; yet, their performance index variance does not diminish when evolution is deactivated. **Fig 13** and **14** together illustrate how iNet facilitates the effective evolution of all agents, resulting in similar and high-performance outputs. **Fig 6** illustrates the extent of self-organization among agents as measured by the entropymetric. The allocation of agents within the goal space serves as the foundation for quantifying entropy. It quantifies the level of chaos among the agents inside the objective space. A decreased entropy shows that agents exhibit more organization. An agent that independently decreases its entropy is deemed self-organizing. Eq. (5) provides a computational method for calculating entropy.

$$\text{Entropy} = -\sum_{i} p_i \times \log(p_i) \tag{5}$$

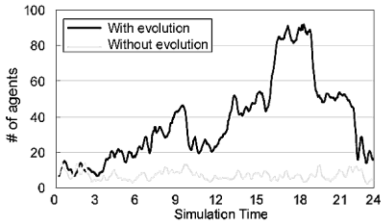

Fig 7. Agent Population

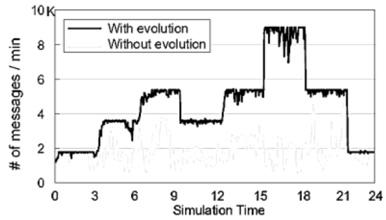

Fig 8. Agent Throughput

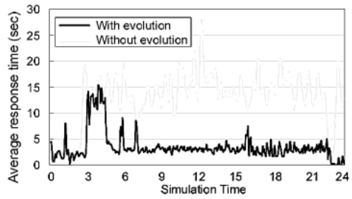

Fig 9. Average Response Time

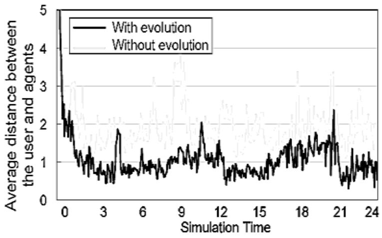

Fig 10. Agent-To-User Average Distance

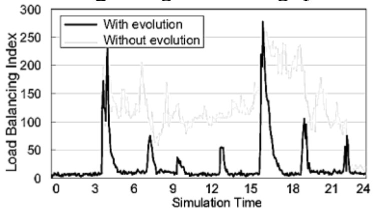

Fig 11. Index for Load Balancing

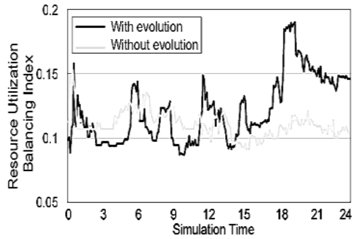

Fig 12. Index for Balancing Resource Utilization

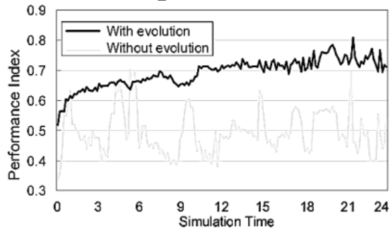

Fig 13. Index of Average Performance

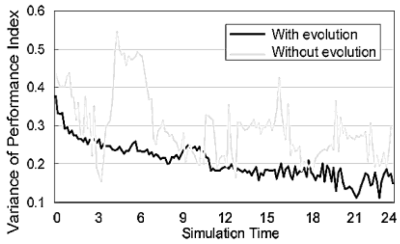

Fig 14. Measure of Performance Variance

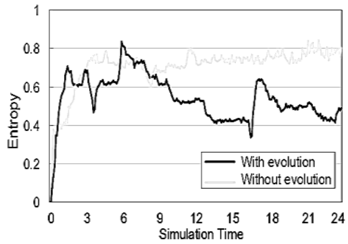

Fig 15. The Level of Self-Organization

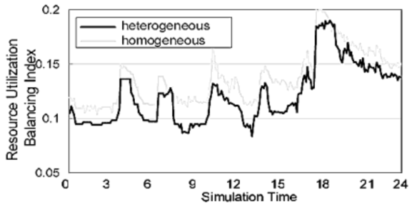

Fig 16. RUBI At Work in Both Homogeneous and Heterogeneous Ensembles

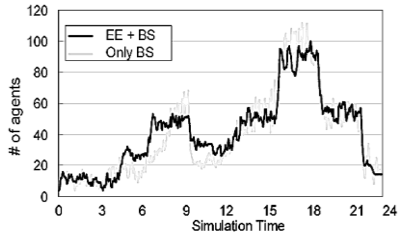

Fig 17. Agents' Population

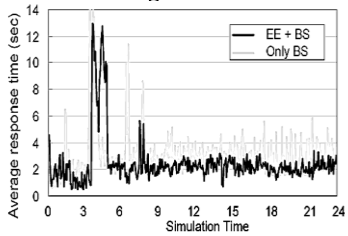

Fig 18. Relative User Response Time

The whole space is partitioned into 27 different cubes, with p_i being the chance that an agent occupies the i^{th} cube. The likelihood is quantified as agents' ratio in the i^{th} cube to the overall number of agents. As seen in **Fig 15**, entropy rises at the start of a simulation due to replicated and duplicated agents producing varying objective figures. Nevertheless, via the evolutionary process, agents progressively reduce their randomness. This indicates that they independently produce comparable objective figures over a significant duration of time. Alongside **Fig 13**, **Fig 15** illustrates that iNet enables all agents to effectively self-organize within objective spaces through evolution, resulting in high-performance outcomes and comparable values. iNet enables agents to adjust to varying environments on hosts, addressing both heterogeneity and homogeneity of resource presence (such as memory capacity). To assess the efficacy of iNet agents in a heterogeneous environment, the simulated model is modeled as a server platform including 50 hosts with memory of 128MB, and 50 hosts with memory of 64MB, arranged in a 10 x 10 grid network topology.

Ouhame, Hadi, and Ullah [17] used a resource utilization method for energy forecasting and to minimize energy consumption and processing duration in resource allocation inside cloud servers. The algorithm improved overall system performance and availability by predicting energy consumption and allocating resources. Resource availability allows for the allocation of virtual machines in cloud servers based on predicted energy load, thereby minimizing energy resource usage and reducing both energy consumption and processing time. In our analysis, **Fig 16** illustrates the trajectory of the RUBI (resource utilization balancing index) in both heterogeneous and homogeneous environments. Agents effectively minimize RUBI to maintain equilibrium in resource usage. In a heterogeneous environment, RUBI diminishes compared to a homogeneous environment due to an increased tendency of agents to engage in high resource migration, namely migrating to nearby hosts with greater resource availability. A greater number of agents operate on hosts comprised of resources.

EE Facility Evaluation

The evolving process of iNet transpires inside the immune network structure at BS. The EE feature also aids in conserving resource usage and reducing performance overhead for behavior choices; moreover, it enhances the flexibility of the agent. In the absence of the EE facility, an agent intermittently monitors environmental parameters and execute one of many actions, irrespective of their adaptability to the surrounding environment. This leads to resource wastage (e.g., memory space and CPU cycles) due to an extraneous behavior selection procedure. In the EE facility, agents first assess their adaptability to the existing environment; they then implement the BS facility only if they fail to adapt.

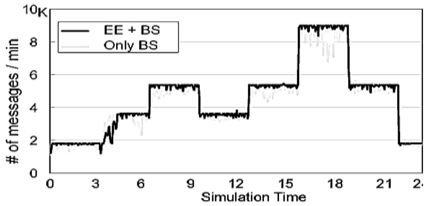

Fig 19. Throughput of Agents

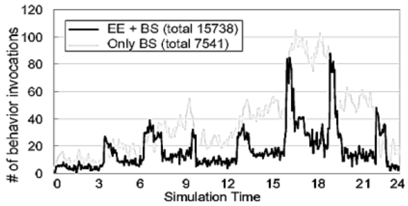

Fig 20. The Amount of Times Behavior Is Invoked

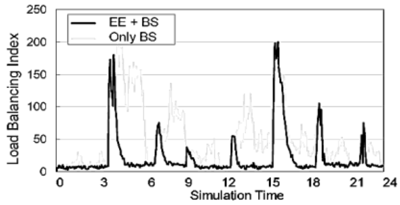

Fig 21. Load Balancing Index

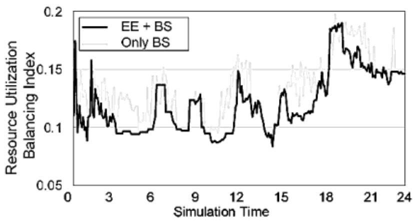

Fig 22. Balanced Index for Resource Utilization

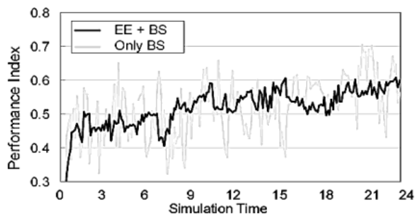

Fig 23. The Mean Agent Performance Index

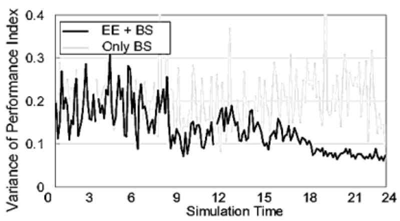

Fig 24. Variability of the Agents' Performance Index

To assess the effect of the EE facility, a comparison is made between two distinct kinds of agents: those with the EE facility (EE plus BS) and those without it (Only BS). Consistent with the prior simulation results (Section IV(A)), the variations in agent population, user response time, agent throughput, LBI, and RUBI are assessed based on the workload drop (refer to **Fig 1**) and illustrated in **Figs 17, 18, 19, 21,** and **22**, correspondingly. Both agent types will significantly enhance their efficiency in comparison to those without evolutionary mechanisms (as detailed in the preceding chapter). However, it has been indicated that an agent without EE facility erroneously exhibit certain actions when such actions are unwarranted. For instance, at approximately 7:00 in **Fig 17**, over 50 agents using the EE facility efficiently handled all requests, totaling roughly 5,499 messages every minute. However, agents without EE capability continuously alter their populations to around 70 by mortality behaviors or replication.

Subsequently, at around 9:00, some of them perish rapidly when the strain diminishes owing to excessive replication and inadequate user energy. Moreover, agents devoid of the EE capacity do not promptly minimize their feedback time for users, since they cannot adjust their population expeditiously (see **Fig 18**). They may relocate considerable distances from the user or expire unpredictably. Correspondingly, there is a delay in performance enhancement between two distinct kinds of agents (i.e., those with/without the EE capacity). Agents using EE facility exhibit more rapid performance enhancement compared to those without it; also, the variability in performance improvement is greater in the absence of the EE facility. The illustrations (gray) for those without EE feature in **Figs 18, 19, 21,** and **22** exhibit instability and oscillation.

Due to superfluous behavior requests, those without EE feature incur extra execution cost and consume resources. **Fig 20** illustrates the cumulative frequency of behavioral requests by active agents in every model cycle, indicating the number of cycles agents execute their actions. For those without EE capability, the quantity of behavior requests precisely corresponds to the number of agents, since they observe environmental circumstances and activate one behavior during every simulation time. Agents inside EE are predisposed to exhibit actions in response to fluctuations in workload. Agents having the EE feature executed BS 7,541 times, but those without it executed it 15,738 times throughout modeling. Agents using the EE capability will reduce behavior invocation frequency by about 47.9%. Although agents using the EE facility exhibit fewer behavioral options, they efficiently enhance their performance index.

Fig 23 illustrates the average agents' performance index without EE is erratic, while **Fig 24** indicates that the agents' variance without the EE feature has not achieved satisfactory convergence. Consequently, the best iNet configuration (genes) is effectively disseminated among other persisting agents; consequently, EE feature also enables agents in adapting to environmental circumstances over generations.

V. DISCUSSION

The research findings demonstrate that iNet evolution process and the Environment Evaluation (EE) facility enhances agent adaptability in alignment with existing studies on adaptive systems and multi-agent networks. The iNet evolution process improves adaptability through its ability to enable agent adjustments of behavior policies, population dynamics and resource utilization thereby surpassing the adaptability of non-evolvable agents. The discovery matches recent research in evolutionary algorithms and self-organizing systems since adaptive mechanisms deliver enhanced system performance during dynamic conditions. The evolutionary capabilities of agents granting them self-directed control of response efficiency together with workforce distribution and resource trajectory demonstrates why adaptive features should be integrated into distributed systems according to findings from Dai et al. [18] regarding swarm intelligence and autonomous agent networks.

Agent adaptability gets improved through EE facility implementation by eliminating needless behavior calls which helps save resources in environments with limited resources. The management approach follows research into cost-effective and energy-efficient resource management practices for cloud computing and distributed systems as described by Guazzone, Anglano, and Canonico [19]. Cloud computing provides utility-based IT services to consumers globally. Utilizing a pay-as-you-go paradigm, it facilitates the hosting of ubiquitous apps across consumer, scientific, and corporate sectors. Data centers that host Cloud applications require significant energy, resulting in elevated operating expenses and substantial carbon footprints. Consequently, we project green cloud computing solutions that can save energy for the environment while also decreasing operating expenses.

Hameed et al. [20] delineate the goal, problems, and architectural components for the energy-efficient administration of Cloud computing infrastructures. Their emphasis was on creating dynamic algorithms for resource provisioning and allocation that account for the interplay of diverse data center infrastructures, including hardware, power units, cooling systems, and software, to enhance overall energy efficiency and performance. They suggest (a) architectural guidelines for cloud energy efficiency; (b) scheduling algorithms and resource allocation policies that take quality-of-service requirements and device power usage characteristics into account; and (c) a new software technology for cloud energy efficiency. Agent performance stability increases with the EE facility because agents first check environmental conditions which reduces execution overhead and scales execution resources. Dynamic resource allocation strategies benefit from intelligent decision systems because they make systems more efficient and resilient according to Zhang and Yang [21].

Through this research the growing literature on self-organizing systems receives input which demonstrates how agents evolve to achieve low entropy and high performance. The entropy metric offers a new method to measure self-organization levels while supporting complex system theories about self-organized systems. The elements or agents in a complex system initially engage only in local interactions, specifically with their close neighbors. The operations of distant agents are initially autonomous, exhibiting no link between activities in different regions. Nonetheless, due to the direct or indirect interconnection of all components, alterations disseminate, resulting in distant areas ultimately being affected by present occurrences. The intricate interaction of positive and negative feedbacks renders this distant impact difficult to forecast and may first seem chaotic. The iNet framework shows its robustness because agents use self-reliant means to achieve matching objective values and balance resource usage across both uniform and diverse environments according to Pulicherla et al. [22].

In recent decades, intelligent and autonomous software agents have increasingly found applications across different fields, including power systems management, flood forecasting, business process management, junction management, and the resolution of complex optimization issues, among others. The essential aspect of comprehending the notion of a multi-agent system (MAS) is intelligent interaction, such as coordination, collaboration, or negotiation. Consequently, MAS are optimally designed to depict issues characterized by many solving approaches, diverse views, and/or the potential for resolution by several actors. Consequently, one of the principal application domains of MAS is extensive computing.

Agents are pivotal in the amalgamation of AI sub-disciplines, often associated with the hybrid architecture of contemporary intelligent systems. Ding et al. [23] addresses a hybrid evolutionary-agent methodology, as indicated by the title. In the majority of analogous applications documented in [24], an evolutionary algorithm is utilized by an agent to facilitate the execution of certain tasks, often associated with learning or reasoning, or to enhance the coordination of group (team) activities. In alternative methods, agents provide a management framework for the distributed implementation of an evolutionary algorithm. The research outcomes emphasize that optimal system performance requires combining evolutionary processes with the intelligent evaluation mechanism which includes the EE facility. The combined method improves agent flexibility while resolving issues connected to resource utilization and program execution overhead in extensive distributed systems. The research extends existing knowledge by developing a complete framework to create adaptive self-organizing systems which work efficiently in unpredictable heterogeneous systems.

VI. CONCLUSION

The research results prove that the combination of agent evolution from iNet and environmental evaluation (EE) creates better adaptive and performant agents for SONs. The research established that evolvable agents show effective adaptation of their population distribution with location placement and behavior policy adjustment to respond to changing workload needs and achieve better throughput and reduced response times with enhanced load-balancing results. Agents equipped with EE technology invoke fewer behaviors unintentionally and this produces improved resource allocation as well as minimizes computational performance impact. Agent evolution and environmental evaluation perform together to resolve SON challenges within heterogeneous and dynamic operational environments according to the research findings. The automatic behavior and policy transformations of agents allows adaptation to network changes which results in better network efficiency ratings. The network implements an approach that ensures high scalability which allows it to manage expanded device and user populations with minimal impact on performance.

CRediT Author Statement
The author reviewed the results and approved the final version of the manuscript.

Data Availability
No data was used to support this study.

Conflicts of Interests
The authors declare no conflict of interest.

Funding
No funding agency is associated with this research.

Competing Interests
There are no competing interests.

References
[1]. A. Petrucci, G. Barone, A. Buonomano, and A. Athienitis, "Modelling of a multi-stage energy management control routine for energy demand forecasting, flexibility, and optimization of smart communities using a Recurrent Neural Network," Energy Conversion and Management, vol. 268, p. 115995, Aug. 2022, doi: 10.1016/j.enconman.2022.115995.
[2]. A. Asghar, H. Farooq, and A. Imran, "Self-Healing in Emerging Cellular Networks: Review, challenges, and research directions," IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 1682–1709, Jan. 2018, doi: 10.1109/comst.2018.2825786.
[3]. J. Yang, Y. Han, Y. Wang, B. Jiang, Z. Lv, and H. Song, "Optimization of real-time traffic network assignment based on IoT data using DBN and clustering model in smart city," Future Generation Computer Systems, vol. 108, pp. 976–986, Dec. 2017, doi: 10.1016/j.future.2017.12.012.
[4]. H. Fourati, R. Maaloul, L. Chaari, and M. Jmaiel, "Comprehensive survey on self-organizing cellular network approaches applied to 5G networks," Computer Networks, vol. 199, p. 108435, Sep. 2021, doi: 10.1016/j.comnet.2021.108435.
[5]. J. Piersa, F. Piekniewski, and T. Schreiber, "Theoretical model for Mesoscopic-Level Scale-Free Self-Organization of functional brain networks," IEEE Transactions on Neural Networks, vol. 21, no. 11, pp. 1747–1758, Oct. 2010, doi: 10.1109/tnn.2010.2066989.
[6]. Z. Song, H. Zhang, and C. Dolan, "Promoting Disaster Resilience: Operation Mechanisms and Self-Organizing Processes of Crowdsourcing," Sustainability, vol. 12, no. 5, p. 1862, Mar. 2020, doi: 10.3390/su12051862.
[7]. K. L. Mills, "A brief survey of self-organization in wireless sensor networks," Wireless Communications and Mobile Computing, vol. 7, no. 7, pp. 823–834, May 2007, doi: 10.1002/wcm.499.
[8]. D. Dhabliya et al., "Energy-Efficient Network Protocols and Resilient Data Transmission Schemes for Wireless Sensor Networks—An Experimental survey," Energies, vol. 15, no. 23, p. 8883, Nov. 2022, doi: 10.3390/en15238883.
[9]. D. Casagrande, M. Sassano, and A. Astolfi, "Hamiltonian-Based Clustering: Algorithms for static and dynamic clustering in data mining and image processing," IEEE Control Systems, vol. 32, no. 4, pp. 74–91, Jul. 2012, doi: 10.1109/mcs.2012.2196321.
[10]. L. Mészáros, A. Varga, and M. Kirsche, "INET Framework," in EAI/Springer Innovations in Communication and Computing, 2019, pp. 55–106. doi: 10.1007/978-3-030-12842-5_2.
[11]. F. Dressler and I. Carreras, Advances in biologically inspired information systems. 2007. doi: 10.1007/978-3-540-72693-7.
[12]. R. Pump, V. Ahlers, and A. Koschel, "Evaluating artificial immune system algorithms for intrusion detection," 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), pp. 92–97, Jul. 2020, doi: 10.1109/worlds450073.2020.9210342.
[13]. E. L. Cooper, "Evolution of immune systems from self/not self to danger to artificial immune systems (AIS)," Physics of Life Reviews, vol. 7, no. 1, pp. 55–78, Dec. 2009, doi: 10.1016/j.plrev.2009.12.001.
[14]. J.-Y. L. Boudec and S. Sarafijanović, "An artificial immune system approach to misbehavior detection in mobile ad hoc networks," in Lecture notes in computer science, 2004, pp. 396–411. doi: 10.1007/978-3-540-27835-1_29.
[15]. P. Degenne et al., "Design of a Domain Specific Language for modelling processes in landscapes," Ecological Modelling, vol. 220, no. 24, pp. 3527–3535, Jul. 2009, doi: 10.1016/j.ecolmodel.2009.06.018.
[16]. P. Vuthi, I. Peters, and J. Sudeikat, "Agent-based modeling (ABM) for urban neighborhood energy systems: literature review and proposal for an all-integrative ABM approach," Energy Informatics, vol. 5, no. S4, Dec. 2022, doi: 10.1186/s42162-022-00247-y.
[17]. S. Ouhame, Y. Hadi, and A. Ullah, "An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model," Neural Computing and Applications, vol. 33, no. 16, pp. 10043–10055, Mar. 2021, doi: 10.1007/s00521-021-05770-9.
[18]. F. Dai, M. Chen, X. Wei, and H. Wang, "Swarm Intelligence-Inspired autonomous flocking control in UAV networks," IEEE Access, vol. 7, pp. 61786–61796, Jan. 2019, doi: 10.1109/access.2019.2916004.
[19]. M. Guazzone, C. Anglano, and M. Canonico, "Energy-Efficient Resource Management for Cloud Computing Infrastructures," 2011 IEEE Third International Conference on Cloud Computing Technology and Science, pp. 424–431, Nov. 2011, doi: 10.1109/cloudcom.2011.63.
[20]. A. Hameed et al., "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems," Computing, vol. 98, no. 7, pp. 751–774, Jun. 2014, doi: 10.1007/s00607-014-0407-8.
[21]. B. Zhang and G. Yang, "Research on the application of intelligent Decision support System in the optimal allocation of higher education resources," 2022 International Conference on Computers, Information Processing and Advanced Education (CIPAE), pp. 656–661, Aug. 2024, doi: 10.1109/cipae64326.2024.00125.
[22]. K. K. Pulicherla, V. Adapa, M. Ghosh, and P. Ingle, "Current efforts on sustainable green growth in the manufacturing sector to complement 'make in India' for making 'self-reliant India,'" Environmental Research, vol. 206, p. 112263, Oct. 2021, doi: 10.1016/j.envres.2021.112263.
[23]. Z. Ding, Z. Sun, R. Liu, and X. Xu, "Evaluating the effects of policies on building construction waste management: a hybrid dynamic approach," Environmental Science and Pollution Research, vol. 30, no. 25, pp. 67378–67397, Apr. 2023, doi: 10.1007/s11356-023-27172-1.
[24]. R. A. Watson, S. G. Ficici, and J. B. Pollack, "Embodied Evolution: Distributing an evolutionary algorithm in a population of robots," Robotics and Autonomous Systems, vol. 39, no. 1, pp. 1–18, Apr. 2002, doi: 10.1016/s0921-8890(02)00170-7.