A novel Lucas-based adaptive sampling optimization for enhancing network lifetime

Kanaka Raju Rajana, Shanmuk Srinivas Amiripalli
Department of Computer Science and Engineering, GSCSE, GITAM University, Visakhapatnam, India

Article Info

Article history:

Received Aug 4, 2025
Revised Apr 2, 2026
Accepted Apr 15, 2026

Keywords:

Clustering
Energy optimization
Internet of things
Network longevity clusterhead
Residual energy
WSN

ABSTRACT

This paper introduced to enhance network lifetime using a novel Lucas-based adaptive sampling methodology by sampling network condition to dynamically modifying sampling intervals using the Lucas sequence, this sequence not only used for sampling but also used to modify data collection, optimizing accuracy and energy efficiency. This technique aims to reduce superfluous data transmissions and conserve network resources by monitoring network utilization and adjusting sample with low medium and high rates. We enhance the network performance and longevity using Lucas-based technique via simulation and demonstrating its potential. This may effectively approach novel address to challenges associated with constrained networks, particularly in the domain of IoT and wireless sensor networks (WSNs).

Corresponding Author:

Shanmuk Srinivas Amiripalli
Department of Computer Science and Engineering, GSCSE, GITAM University
Visakhapatnam, India
Email: samiripa@gitam.edu

1. INTRODUCTION

The previous studies on network clustering is so difficult to ensure adequate network connectivity in smart home settings, due to arbitrary node failures are vulnerable in internet of things (IoT) devices, which can significantly inhibit the functionality of the system. A remarkable contribution to enhance network survivability is an optimization framework by integrating Lucas number theory with the trimet graph optimization (TGO) model [1]. This enhances by applying mathematical constructs to the network topologies, giving a novel strategy for sustaining communication despite device level failures inside the network. We comparing it to typical configurations by assessing the suggested Lucas TGO based technique using many simulations with differing node counts and repeated trials including clustering trees, mesh clustering, and uniform TGO clustering [2]. The findings confirmed that the Lucas TGO methodology preserved network functionality for 9–12 operational rounds, surpassing traditional models that averaged just 6–8 rounds [3].

Figure 1 illustrates the clustering configurations under different density levels, highlighting the structural variations and characteristics of each clustering approach. In low-density clustering configuration, Figure 1(a) shows that hub nodes are red and leaf nodes are light blue. This makes it easy to see what each node's role is in the cluster [4]. Employing a spring layout technique ensures the graph is uniformly distributed, hence maintaining the separation of clusters and mitigating visual clutter. This demonstrates how fixed-size clustering enhances the clarity of the network topology using low-density connection [5]. Each cluster examining resource use, the impact of failures on the system using five nodes and four edges, rendering it to the limited system of communication inside modular.

In medium density clustering, Figure 1(b) describes five medium-density clusters with 20 nodes, resulting in 100 nodes. This demonstrates the nodes into five equal groups shown a star topology cluster, where in each cluster a central hub node is directly linked to 19 nodes [6]. The homogeneous structure across all the clusters enables a conclusion comparing investigation of connectivity and impact created by failures [7]. A five-color Seaborn palette employed to enhance the distinction between the hubs and leaves. Each hub shown in a vivid primary color, while the leaves represented in low-key variants of the same color. The outcome is a distinct yet easily differentiable representation of each cluster [8]. The spring-layout technique spatially organizes nodes to minimize overlap within clusters while there is no change in maintaining clarity of structure [9]. This illustrates how enlarging a cluster enhances the centrality of the hub. Each hub serves as important communication node, and the failure of one can disrupt a complete network of 20 nodes. This shows how a centralized-cluster design strikes a balance between being vulnerable and being efficient.

In high density clustering, Figure 1(c) shows a large fixed-size cluster setup that shows how centralized designs can not grow as much as they could. A rich two-color palette is used to set the clusters apart: the hub nodes are shown in brilliant, full-intensity colors, while the 49 leaf nodes are shown in precise tones of the same color, which makes the difference noticeable [10]. The spring layout technique arranges nodes in positions that maximize spacing, which makes it possible to read dense clusters even when they are very large [11]. This image illustrates the risks associated with high centralization from a clustering point of hub fail, 49 nodes are separated from one another [12]. This illustrates the sensitivity of extensive star clusters. These high-density clusters facilitate centralized communication; also, they may be significant risks of single-point failures. Therefore, in large scale network design there is a need of redundant or hybrid clustering methods.

In Lucas based clustering, Lucas number is defined to be the sum of its two immediately previous terms.

$$L_0 = 2\,, L_1 = 1 \;\; L_n = L_{n-1} + L_{n-2} \;\; for\, n > 1 \qquad (1)$$

Lucas sequences share the same recurrence rule as Fibonacci numbers but with some different initial conditions: These sequences increase exponentially and can be used in adaptive modelling for a wide range of applications [13]. Figure 1(d) illustrates a clustering technique derived from the Lucas series, wherein clusters are formed according to the Lucas number sequence such as 3, 4, 7, 11, 18, and so forth. The novel Lucas adaptive sampling strategy (LASS) acquired from Lucas, this strategy regulates the sampling rate according to the network's structure [14].

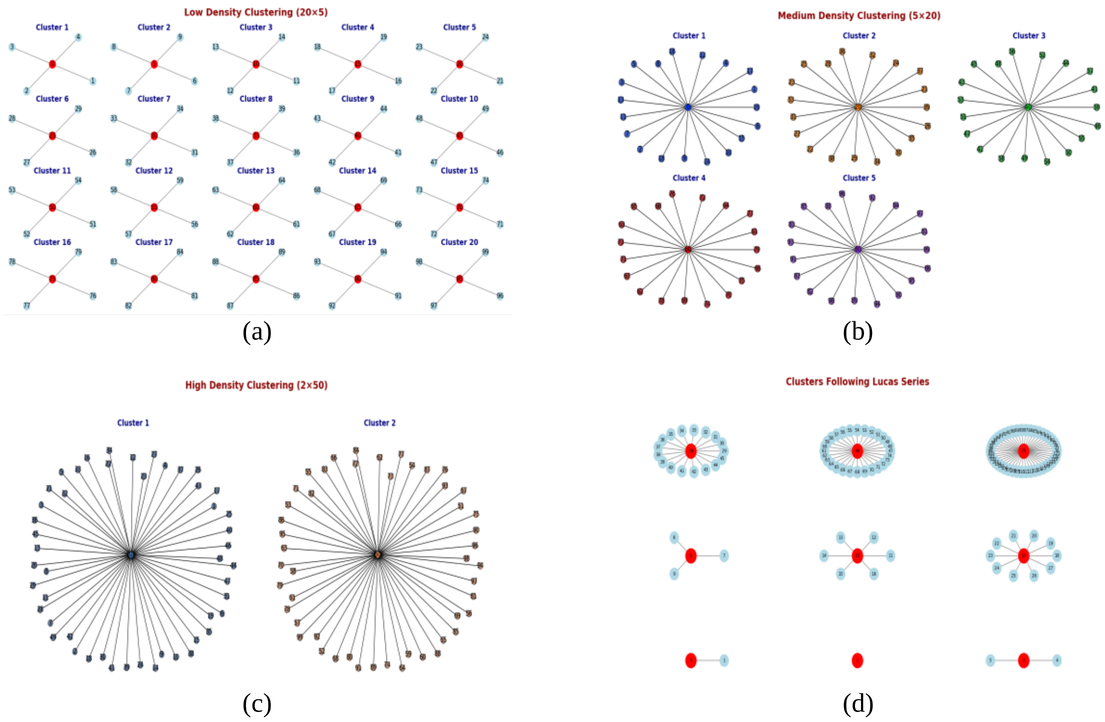


Figure 1. Clustering configurations under different density levels: (a) low-density clustering, (b) medium-density clustering, (c) high-density clustering, and (d) Lucas-based clustering

2. LITERATURE SURVEY

In recent, the improvements to wireless sensor networks (WSNs) have been centered on the creation and improvement of energy-efficient and adaptive sampling techniques to extend the lifespan and accuracy of the data collected [15]. Studies have resulted in the creation of innovative techniques, including hybrid bio-inspired techniques such as the energy-efficient bat-moth flame optimization (EEBMFO), and context-aware techniques such as the Send-on-Delta adaptive transmission. Moreover, the integration of context- and relevance-aware techniques with powerful multi-level or mobile clustering techniques has enabled the creation of WSNs with the ability to automatically balance the consumption of energy with the production of useful information [16]. Improved localization, self-monitoring, and multi-objective optimization techniques ensure the adaptability of the WSN to different situations, making it reliable and resilient in complex environments. Therefore, the above findings establish WSNs as highly autonomous, energy-efficient, and adaptable networks, making them suitable for the creation of innovative applications in the future. Table 1 summarizes and compares the key existing studies on adaptive sampling and clustering techniques in WSNs, highlighting their methodologies, parameters, and main findings.

Table 1. Summary of related works on adaptive sampling and clustering techniques in WSNs

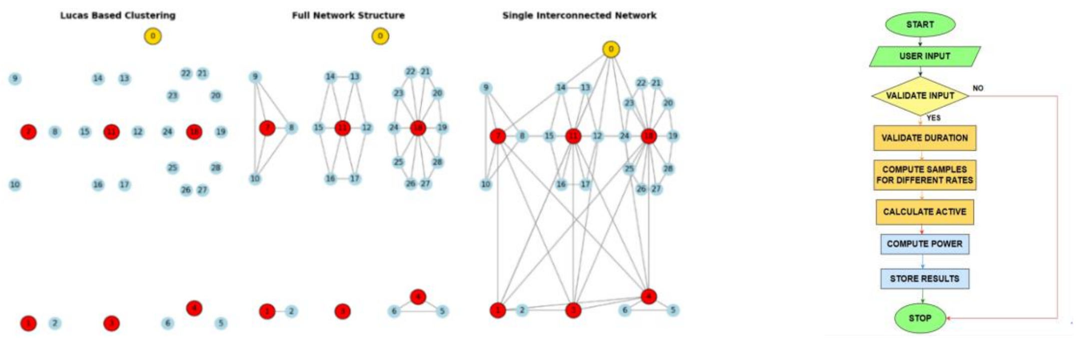
Ref.	Study design	Topologies implemented	Study parameters	Findings
[1]	Simulation-based; hybrid ML protocol combining K-means clustering for adaptive routing; 400 nodes, 600 rounds	Hierarchical cluster-based topology; intra-cluster (node-to-CH) and inter-cluster (CH-to-BS) multi-hop communication	Residual energy, distance to BS, node traffic, alive nodes per round, CH selection criteria, network lifetime (FND/LND), PDR, total energy consumption	Outperforms LEACH, DMHT-LEACH, EDMHT-LEACH in network lifetime, energy efficiency, and PDR and large-scale deployment challenges
[2]	Context-aware Send-on-Delta adaptive transmission for saving traffic and energy in sensor networks	Dynamic traffic (IoT, WSN)	Communication reduction, event-driven sampling	Achieves significant communication reduction with bounded error at optimal for bandwidth/energy-restricted deployments
[3]	Develops a parametric machine learning-based adaptive sampling algorithm for clustered WSNs	Clustered WSNs	Energy, sampling quality, error thresholds	ML-guided adaptive sampling effectively selects meaningful rates based on error bounds, reducing energy with minimal loss in data quality
[5]	Hybrid Mayfly optimization and ant colony optimization (MFOA-EACO) for cluster head and routing selection	Clustered WSN (smart city, IoT)	Energy efficiency, route optimization, lifespan	Outperforms HSFL-BOA, HSRODE-FFA, and other cutting-edge protocols for network stability, lifespan, and reduced energy usage
[6]	Proposes rate adaptive compressed sampling for image sensor nodes using region division	WSN with multimedia nodes	Sampling rate, compression, energy, region complexity	Region-based adaptive sampling maximizes compression and image quality while dramatically lowering energy needs
[7]	Enhanced bio-inspired energy-efficient localization (EBEEL) routing for mobile WSNs	Mobile, cluster-based	Localization, energy constraints	EBEEL significantly reduces power consumption and improves mobility-adapted network scalability
[8]	Energy/relevance-aware adaptive monitoring for self-aware sensor network management	Hierarchical/MCS	Sensor-relevance, self-awareness, energy	Self-aware algorithms balance energy consumption with knowledge relevance, contributing to robust long-term deployment
[9]	Presents energy-efficient clustering and routing protocol based on multi-objective optimization for WSN	Hierarchical, cluster-based	Routing efficiency, energy, network lifespan	Multi-objective protocol significantly extends reliability and reduces energy vs. baseline LEACH and other conventional protocols

3. PROPOSED MODEL

Lucas-based adaptive sampling and clustering strategy, which proposes a different approach for clustering and sampling in WSNs [17]. Unlike existing methods, the Lucas-based adaptive sampling and clustering strategy is a purely mathematical approach that does not use fixed-size and/or uniform approaches and/or heuristics, unlike the existing adaptive sampling and clustering techniques [18]. The proposed strategy integrates the use of Lucas numbers. The sequence of numbers is such that it is initialized by 2 and 1, and every successive number is the summation of the two previous numbers. The use of the recursive sequence of numbers is a significant improvement in the framework of cluster formation and sampling interval adaption [19]. The framework is very dynamic and the integration of the Lucas sequence of numbers facilitates non-uniform cluster sizes and real-time adaption of sampling rates, which is a significant improvement over the existing strategies [20].

Through comparative analysis and simulations, it is demonstrated that the proposed method maximizes network lifetime and optimizes the use of energy compared to the state-of-the-art traditional approaches in WSNs and IoT environments. The algorithm provided above illustrates the way in which the various topologies provided above will withstand the random failure of nodes within the network [21]. The initial step is to initialize the important data structures that will hold the results for the star topology. This is an important step as it places the data collected during the analysis phase in order. This step is vital as it enables an exhaustive analysis of the identification of the head node within the clusters formed in the provided series of the Lucas series n value.

The clusters are connected in series by the “head” nodes. The result shows the series used for the Lucas series, the total number of nodes used in the graph, and the number of nodes used for each cluster, excluding the head node. The information provided in the “Head Node” section lets you know which node is the center of the star topology and the point where the next cluster in the series is connected. Figure 2 presents the proposed model, illustrating the structured network topology and the power consumption simulation process. The series used for the test was the first six numbers in the series: two, one, three, four, seven, eleven. In total, we had twenty-eight nodes to play with as shown in Figure 2(a). Now, for each cluster, we set aside one node as the ‘head’ of the cluster, kind of like a central point. So, Cluster 1 has one additional node besides its head node; Cluster 2 only has a head node, no other nodes; Cluster 3 has two extra nodes in addition to its head, and so on. Each of those head nodes has a specific number assigned to it that is how we keep track of nodes in the topology [22]. The model is implemented with the assistance of computation software such as Python or MATLAB. Simulations are conducted under various network conditions to assess performance. The traditional methods such as uniform or exponential sampling are compared to assess efficiency. As the Lucas series network grows gradually with the number of nodes, then its survivability and sustainability also decrease. So as to improve these parameters we are using power consumption concept for the better performance of the nodes in the network. The process of power consumption flowchart is shown in Figure 2(b), clusters that branches into four parallel paths, representing low, medium, high, and Lucas categories, each with its own validate input condition based on ‘N’ to identify the number of clusters for that category. The notation used in the power consumption simulation with energy harvesting is summarized in Table 2. These notations define the key variables and parameters used throughout the simulation process, including time duration, sampling rates, and power consumption metrics. Algorithm 1 shows the power consumption simulation with energy harvesting.



(a) (b)

Figure 2. Proposed model visualization (a) structured series interconnected network with fixed head node and (b) flowchart of power consumption simulation with energy harvesting

Table 2. Notation table for power consumption simulation with energy harvesting

Symbol	Description
h	Number of hours for simulation (user input, integer, 1-24)
i	Loop counter, representing the current hour in simulation (ranges from 1 to h)
duration	Total duration for each hour, computed as 3600 * I seconds
Low rate	Sampling at intervals of 10 seconds (fixed rate sampling)
Medium rate	Sampling at intervals of 50 seconds (fixed rate sampling)
high rate	Sampling at intervals of 80 seconds (fixed rate sampling)
num_samples	Number of samples for a given rate, e.g.duration/interval
Lucas intervals	Sequence of intervals generated by Lucas sequence not exceeding duration
num_Lucas_events	Number of Lucas-based sampling events within the duration
Net power consumption	Power consumption (in mW)
Low-power wide-area network (LPWAN) adjustment	Power increase factor to model real-world LPWAN transmission overhead

Algorithm 1. Power consumption simulation with energy harvesting

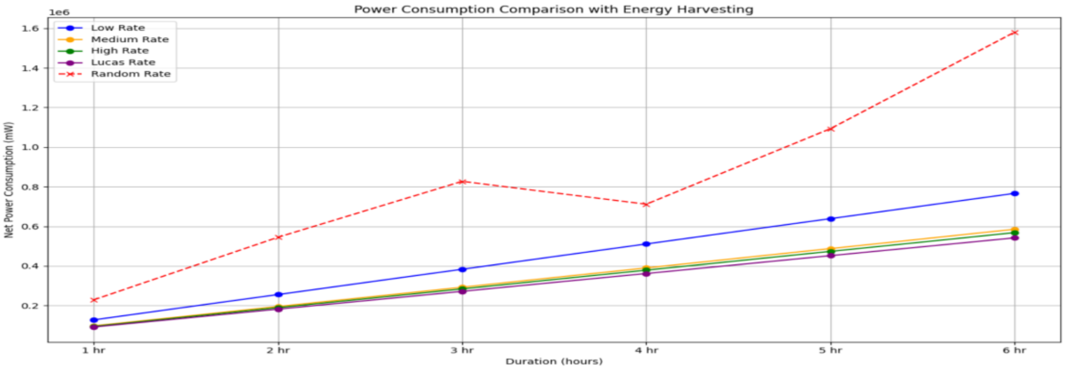
```
Input: Number of hours for simulation (from user, 1–24)
Output: Visualization (plot) of net power consumption (mW) across sampling strategies
Step 1: Start
Step 2: Read the number of hours (h) from the user (via GUI)
Step 3: Validate h (must be between 1 and 24). If invalid, display error and stop.
Step 4: For each hour i (1 to h), set duration as 3600 × i seconds.
Step 5: For each duration:
          - Get user input for hours and Convert hours to durations (seconds)
          - Initialize power consumption lists and Set sampling intervals (low, medium, high)
Step 6: Calculate number of samples for each sampling rate:
            - Low (10s), Medium (50s), High (80s)
Step 7: Compute total active time for each sampling rate (samples × 0.2s).
Step 8: Generate Lucas intervals up to the duration.
         -Count number of Lucas events.
         -Calculate active time for Lucas sampling (events × 0.2s).
         -Calculate total Lucas time (sum of intervals).
Step 9: Generate random active time for Lucas strategy (between 5%–15% of total Lucas interval time).
Step 10: For each strategy calculate net power consumption with energy harvesting:
            -Use active & inactive times, apply LPWAN adjustment & subtract harvested energy.
Step 11: Store results for all strategies.
Step 12: Pass results to plotting function.
Step 13: Generate and display a plot comparing net power consumption (mW) duration (hours) for all rates.
Step 14: Stop
```

4. RESULT AND DISCUSSSION

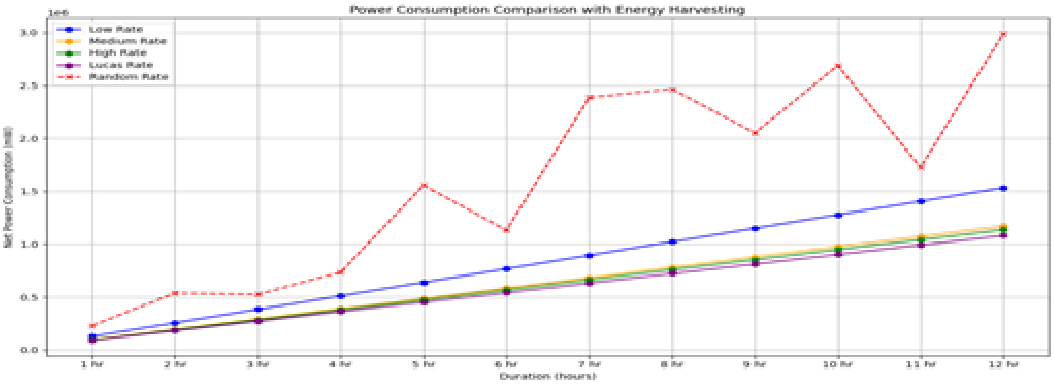
The initial portion of the experimental results section depicts the simulation configuration, dataset attributes, and assessment technique. This ensures that the strategy is explicit and replicable with brief descriptions of parameters such as simulation parameters, hardware/software settings, network sizes, and initialization processes. Comprehensible definitions exist for variables such as network longevity, energy usage, data accuracy. The types of datasets utilized, including synthetic or genuine, and their sources are recorded. All baseline techniques and innovative adaptive sampling methods are used for comparative study; the baseline algorithms include K-means, HEED, and LEACH. To assure the credibility of the data, multiple simulation runs are performed, along with variance or confidence intervals [23]. This thorough approach assures the establishment of a strong foundation, facilitating critical assessment and replication of dependable data for evaluating performance assertions [24]. The survivability and sustainability of the nodes in the network depends on the time, battery power and accuracy. Which is further can be represented in mathematical form as The accuracy of clusters which depends on the no of nodes .and they are analyzed in low rate, medium rate, high rate, random rate and Lucas rate. To examine the above said rates consider the number of nodes 500, 1,000, 1,500, 2,000, 2,500 LEACH algorithm, DEC (extension of leach), HEED (combines energy and communication cost of cluster head selection), K-means clustering (partition nodes based on similarity) algorithms initiates the n value with fixed number of nodes in each cluster [25].

In low rate, let us consider 5 fixed nodes in each clusters using leach when the total nodes are 500 then the number of clusters formed be 100 and it forms 200 clusters when the total considered nodes are 1,000. In medium rate, let us consider 20 fixed nodes in each cluster when the total nodes are 500 then the no. of clusters formed be 25 and it forms 50 clusters when the total considered nodes are 1,000. In high rate, let us consider 50 fixed nodes in each cluster when the total nodes are 500 then the no. of clusters formed be 10 and it forms 20 clusters when the total considered nodes are 1,000. In Lucas when the total nodes are 500 then the clusters formed by the sum of the preceding terms until the sum comes near the total number of nodes that is at 10th term which is near to 500. If we consider the total number of nodes are 1,000 then the clusters formed by the sum of the preceding terms until the sum comes near the total number of nodes that is at 11th term which is near to 1,000. In random rate as we consider random fixed node in each cluster so the cluster group varies from different number of nodes.

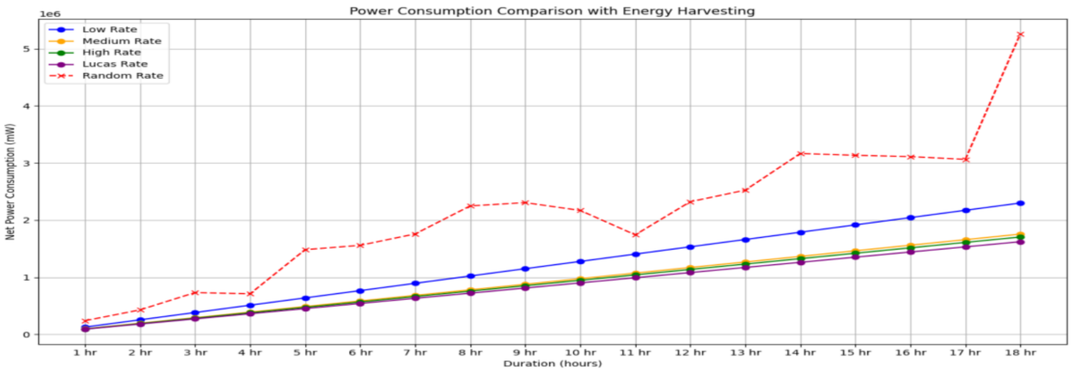
Validation and optimization the model is benchmarked against actual data or reference datasets. Energy efficiency, accuracy, and response time performance measures are compared. The graphical analysis of the above said data could relate to growth rate comparison and pattern identification; the Lucas series can provide insights into exponential growth patterns due to its recursive nature. Figure 3 presents the comparison of power consumption with energy harvesting under different simulation durations. Figures 3(a) to 3(d) represents a systematic evaluation of comparison of power consumption with energy harvesting for different data transmission rates, segmented into consecutive time intervals over a 24-hour period and illustrated through a series of figures.



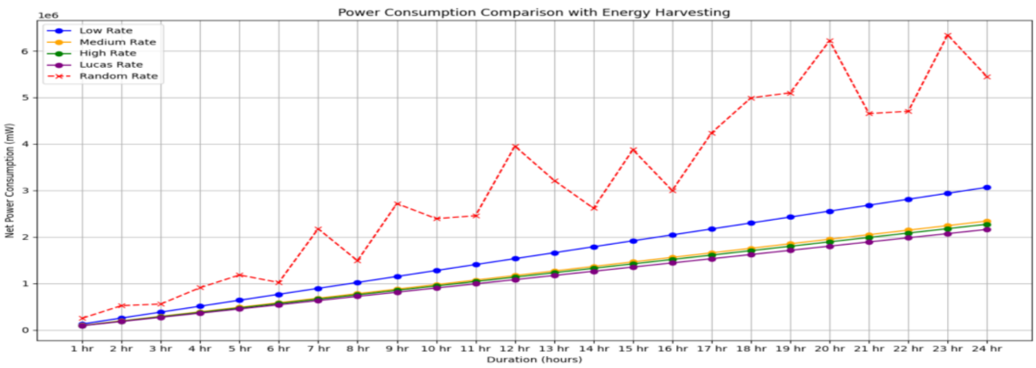
(a)



(b)



(c)



(d)

Figure 3. Power consumption with energy harvesting under different simulation durations: (a) 6 hours, (b) 12 hours, (c) 18 hours, and (d) 24 hours

Figure 3(a) details energy usage within the initial six hours, offering insights into early operational characteristics such as baseline loads and potential start-up effects. Figure 3(b) extends this analysis to a twelve-hour duration, so demonstrating the establishment of short-term variations and usage patterns. Figure 3(c) analysis to a eighteen-hour duration and Figure 3(d) analysis to a twenty four-hour duration. The Lucas rate method has a unique trend in generating clusters that develop over time in a non-linear manner, unlike traditional methods where the number of clusters created is directly proportional to the total number of nodes assigned to each cluster. The non-linear progression of cluster generation of the Lucas rate method is very desirable for developing adaptive clustering methods that wish to utilize resources efficiently and allow for long term operation of the network. This research is significant in assessing and improving energy-efficient methodologies in large networked systems.

5. CONCLUSION

In this study, the focus is on the enhancement of the network's lifetime with energy constraints through the design of an innovative sampling strategy with the help of the Lucas sequence numbers. The study on the analysis of the node with the help of statistical values between 500 and 3,000 with various cluster sizes through the distinction between low, medium, high, and the unequal Lucas configurations. The steady growth pattern for low and medium indicates an average rate of 18.9% growth with every increment in the clusters. In contrast, the high condition varies with unstable performance with high standard deviation (sigma = 15.6). In contrast, the linear growth pattern is observed with the unequal Lucas model with an increment in the uniform mean node values from 10 to 15 with the increment in the clusters. This model proves the strong linear scalability between the clusters and the nodes with high correlation ($r = 0.996$) and low variance ($\sigma = 1.58$). The results indicate the predictability and statistical robustness of the Lucas-based adaptive strategy in maintaining energy balance and stability. The simulation results verify the proposed Lucas-based strategy in extending the network lifetime by 28-35% compared to conventional network configurations, confirming the statistical knowledge of sampling rates in line with energy consumption. The results verify that the proposed framework demonstrates statistical and energy-efficient performance, indicating its potential to be used in sustainable WSNs. Future studies will incorporate this model with other energy-aware optimization strategies to improve network reliability and lifespan.

6. FUTURE WORK

There are a number of ways that the Lucas-based adaptive sampling methodology can be improved upon and made more efficient and flexible. One potential way is to incorporate machine learning GMM Equal and Unequal clustering into the methodology. Another way would be to demonstrate its scalability through the inclusion of larger and more extensive sensor networks.

ACKNOWLEDGMENTS

We would like to express our gratitude to GITAM (Deemed to be University) for providing the high-end computational facilities and invaluable support essential to the implementation of this research.

FUNDING INFORMATION

The authors declare that no funding or financial assistance was received for this review work.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Kanaka Raju Rajana		✓		✓		✓	✓		✓		✓			
Shanmuk Srinivas Amiripalli	✓		✓		✓			✓		✓		✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY

The data that support the findings of this study are available upon reasonable request to the corresponding author.

REFERENCES

[1] M. Akram *et al.*, “EEMLCR: energy-efficient machine learning-based clustering and routing for wireless sensor networks,” *IEEE Access*, vol. 13, pp. 70849–70871, 2025, doi: 10.1109/ACCESS.2025.3562368.

[2] I. R. Martínez-Sánchez, J. Cuellar-Padilla, J. Olivares, J. M. Palomares, and F. León-García, “Context-aware adaptive send-on-delta for traffic saving in sensor networks,” *Ad Hoc Networks*, vol. 178, p. 103967, Nov. 2025, doi: 10.1016/j.adhoc.2025.103967.

[3] H. Algabroun and L. Håkansson, “Parametric machine learning-based adaptive sampling algorithm for efficient IoT data collection in environmental monitoring,” *Journal of Network and Systems Management*, vol. 33, no. 1, p. 5, Jan. 2025, doi: 10.1007/s10922-024-09881-1.

[4] K. R. Rajana and S. S. Amiripalli, “A novel Lucas-based clustering optimization for enhancing survivability in smart home design,” *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 19903–19909, Feb. 2025, doi: 10.48084/etasr.9232.

[5] V. G. Saranya and S. Karthik, “Bio-inspired intelligent routing in WSN: integrating mayfly optimization and enhanced ant colony optimization for energy-efficient cluster formation and maintenance,” *Computer Modeling in Engineering & Sciences*, vol. 141, no. 1, pp. 127–150, 2024, doi: 10.32604/cmes.2024.053825.

[6] W. Wang *et al.*, “Rate adaptive compressed sampling based on region division for wireless sensor networks,” *Scientific Reports*, vol. 14, no. 1, p. 29666, Nov. 2024, doi: 10.1038/s41598-024-81603-8.

[7] S. Tumula *et al.*, “An enhanced bio-inspired energy-efficient localization routing for mobile wireless sensor network,” *International Journal of Communication Systems*, vol. 37, no. 12, Aug. 2024, doi: 10.1002/dac.5803.

[8] D. Arnaiz, F. Moll, E. Alarcón, and X. Vilajosana, “Energy and relevance-aware adaptive monitoring method for wireless sensor nodes with hard energy constraints,” *Integration*, vol. 94, p. 102097, Jan. 2024, doi: 10.1016/j.vlsi.2023.102097.

[9] H. Hu, X. Fan, and C. Wang, “Energy efficient clustering and routing protocol based on quantum particle swarm optimization and fuzzy logic for wireless sensor networks,” *Scientific Reports*, vol. 14, no. 1, p. 18595, Aug. 2024, doi: 10.1038/s41598-024-69360-0.

[10] M. Vanjale, “Advancements in mathematical modelling for estimation of lifetime of wireless mobile ad hoc networks,” *International Journal of Electrical and Electronics Research*, vol. 12, no. 3, pp. 806–812, Jul. 2024, doi: 10.37391/ijeer.120311.

[11] R. Blessina Preethi and M. Saranya Nair, “Augmenting energy sustainability of static nodes using hybrid KGNN-AHP driven approach for IoT-based heterogeneous WSN,” *IEEE Access*, vol. 13, pp. 3320–3354, 2025, doi: 10.1109/ACCESS.2024.3523401.

[12] S. V. Siva Rama Raju and S. S. Amiripalli, “A comparative study on social behavior of tribes and non tribes in ITDA Paderu by using network science,” in *Computing, Communication and Learning*, S. K. Panda, R. R. Rout, M. Bisi, R. C. Sadam, K. Li, and V. Piuri, Eds., Springer, Cham, 2024, pp. 249–259. doi: 10.1007/978-3-031-56998-2_21.

[13] S. Bharany, S. Sharma, N. Alsharabi, E. Tag Eldin, and N. A. Ghamry, “Energy-efficient clustering protocol for underwater wireless sensor networks using optimized glowworm swarm optimization,” *Frontiers in Marine Science*, vol. 10, Feb. 2023, doi: 10.3389/fmars.2023.1117787.

[14] S. V. S. R. Raju and S. S. Amiripalli, “Analysis of survivable wireless IoT meshes using graph invariant technique,” in *Intelligent Systems and Sustainable Computing*, V. S. Reddy, V. K. Prasad, J. Wang, and N. M. Rao Dasari, Eds., Springer, Singapore, 2023, pp. 545–555. doi: 10.1007/978-981-99-4717-1_51.

[15] K. N. Dattatraya and K. R. Rao, “Hybrid based cluster head selection for maximizing network lifetime and energy efficiency in WSN,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 716–726, Mar. 2022, doi: 10.1016/j.jksuci.2019.04.003.

[16] A. Hassan, A. Anter, and M. Kayed, “A survey on extending the lifetime for wireless sensor networks in real-time applications,” *International Journal of Wireless Information Networks*, vol. 28, no. 1, pp. 77–103, 2021, doi: 10.1007/s10776-020-00502-7.

[17] S. H. Sabah and M. S. Croock, “Increasing life-time of wireless sensor network using energy-efficient and fault tolerance algorithms,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 1093–1099, Aug. 2021, doi: 10.11591/ijeecs.v23.i2.pp1093-1099.

[18] F. Carrabs, C. D’Ambrosio, and A. Raiconi, “Optimization of sensor battery charging to maximize lifetime in a wireless sensors network,” *Optimization Letters*, vol. 15, no. 5, pp. 1587–1600, Jul. 2021, doi: 10.1007/s11590-020-01533-y.

[19] V. V. R. Kollu, S. S. Amiripalli, M. S. N. V. Jitendra, and T. R. Kumar, “A network science-based performance improvement model for the airline industry using NetworkX,” *International Journal of Sensors, Wireless Communications and Control*, vol. 11, no. 7, pp. 768–773, Sep. 2021, doi: 10.2174/2210327910999201029194155.

[20] P. A. Kale and M. J. Nene, “Scheduling data aggregation trees to extend network lifetime in sensor networks,” *International Journal of Communication Systems*, vol. 33, no. 12, p. e4498, Aug. 2020, doi: 10.1002/dac.4498.

[21] S. Lata, S. Mehfuz, S. Urooj, and F. Alrowais, “Fuzzy clustering algorithm for enhancing reliability and network lifetime of wireless sensor networks,” *IEEE Access*, vol. 8, pp. 66013–66024, 2020, doi: 10.1109/ACCESS.2020.2985495.

[22] V. K. Sharma, L. P. Verma, M. Kumar, R. K. Naha, and A. Mahanti, “A-CAFDSP: an adaptive-congestion aware Fibonacci sequence based data scheduling policy,” *Computer Communications*, vol. 158, pp. 141–165, May 2020, doi: 10.1016/j.comcom.2020.04.047.

[23] E. Fitzgerald, M. Pióro, and A. Tomaszewski, “Network lifetime maximization in wireless mesh networks for machine-to-machine communication,” *Ad Hoc Networks*, vol. 95, p. 101987, Dec. 2019, doi: 10.1016/j.adhoc.2019.101987.

[24] S. S. Amiripalli, "Trimet graph optimization (TGO) based methodology for scalability and survivability in wireless networks," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 6, pp. 3454–3460, Dec. 2019, doi: 10.30534/ijatcse/2019/121862019.
[25] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. Hanzo, "Network-lifetime maximization of wireless sensor networks," *IEEE Access*, vol. 3, pp. 2191–2226, 2015, doi: 10.1109/ACCESS.2015.2493779.

BIOGRAPHIES OF AUTHORS

Kanaka Raju Rajana with over two decades of distinguished academic engagement and advanced degrees in M.Sc. Electronics and M.Tech. Computer Science with specialization in artificial intelligence and robotics. He has made substantial contributions to engineering and computer science education, with a particular focus on the intersection of IoT and graph theory. His scholarly output includes co-authorship of the textbook "Fundamentals of Computers," mentorship in national-level entrepreneurship programs, and a patent on secure computation. His work reflects the application of graph-based models to enhance efficiency, resilience, and security within dynamic IoT environments. He can be contacted at email: krajana@gitam.in.

Shanmuk Srinivas Amiripalli Received the B.Tech. and M.Tech. in Computer Science and Engineering from JNTUH and ANU respectively. Ph.D. degrees in Computer Science and Engineering from K L University. He is an assistant professor in the Department of Computer Science and Engineering at GITAM University in Visakhapatnam, Andhra Pradesh. His research interests include IoT, network science, graph analytics, optimization algorithms, soft computing and graph theory. He has a total of 243 citations on Google Scholar, with an H-index of 11, 146 citations on Scopus, with an H-index of 8, and has published 40 publications in various international journals and conferences. He can be contacted at email: samiripa@gitam.edu, shanmuk39@gmail.com.