SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ DERGİSİ

Sakarya University Journal of Science
SAUJS

ISSN 1301-4048 e-ISSN 2147-835X Period Bimonthly Founded 1997 Publisher Sakarya University
http://www.saujs.sakarya.edu.tr/

Title: Machine Learning Supported Nano-Router Localization in WNSNs

Authors: Ömer GÜLEÇ

Recieved: 2023-02-02 00:00:00

Accepted: 2023-03-06 00:00:00

Article Type: Research Article

Volume: 27
Issue: 3
Month: June
Year: 2023
Pages: 590-602

How to cite
Ömer GÜLEÇ; (2023), Machine Learning Supported Nano-Router Localization in WNSNs. Sakarya University Journal of Science, 27(3), 590-602, DOI: 10.16984/saufenbilder.1246617
Access link
https://dergipark.org.tr/en/pub/saufenbilder/issue/78131/1246617

Sakarya University Journal of Science 27(3), 590-602, 2023

Machine Learning Supported Nano-Router Localization in WNSNs

Omer GULEC *1

Abstract

Sensing data from the environment is a basic process for the nano-sensors on the network. This sensitive data need to be transmitted to the base station for data processing. In Wireless Nano-Sensor Networks (WNSNs), nano-routers undertake the task of gathering data from the nano-sensors and transmitting it to the nano-gateways. When the number of nano-routers is not enough on the network, the data need to be transmitted by multi-hop routing. Therefore, there should be more nano-routers placed on the network for efficient direct data transmission to avoid multi-hop routing problems such as high energy consumption and network traffic. In this paper, a machine learning-supported nano-router localization algorithm for WNSNs is proposed. The algorithm aims to predict the number of required nano-routers depending on the network size for the maximum node coverage in order to ensure direct data transmission by estimating the best virtual coordinates of these nano-routers. According to the results, the proposed algorithm successfully places required nano-routers to the best virtual coordinates on the network which increases the node coverage by up to 98.03% on average and provides high accuracy for efficient direct data transmission.

Keywords: Wireless nano-sensor networks, IoNT, machine learning, nano-router localization

1. INTRODUCTION

Wireless Nano-Sensor Networks (WNSNs) consist of a large group of energy-limited nano-machines running in the Terahertz (THz) band which are distributedly placed on the nano-domain used in smart healthcare, environmental monitoring, robotics, food industry, security, medicine, military and agriculture applications [1-3].

In WNSN architecture, the network is formed by the nano-devices. These devices are tiny machines that are running on WNSNs for Internet of Nano-Things (IoNT) applications [4]. To be more specific, they are divided into sub-categories according to their types. These devices are called nano-sensors, nano-routers, nano-interfaces and nano-gateways [5].

The basic nano-device on the network is the nano-sensor that is responsible for sensing the data from the environment at the molecular level such as temperature, pressure, glucose concentration, chemicals, oxygen levels, heartbeats and such values [3].

In a WNSN, the data sensed from the nano-sensors need to be transmitted eventually to the base station for data processing. In

* Corresponding author: omer@pau.edu.tr (O. GULEC)
[1]Pamukkale University
ORCID: https://orcid.org/0000-0002-7657-3230

WNSNs, this task is undertaken by the nano-routers. A nano-router transmits the data from nano-nodes to the nano-interfaces. The nano-interface also transmits data from nano-routers to nano-gateways.

A nano-gateway is a nano-device that gathers data from nano-routers on a WNSN and sends it to the base station for data processing over the Internet. The nano-gateways are typical devices such as wristbands, wound bands, smartwatches and smartphones.

The illustration of hierarchical WNSN architecture running on IoNT applications is given in Figure 1. In an example WNSN shown in the figure, there are 2 nano-routers placed on the network that gather data from the nano-sensors which are represented by green nodes. These nano-routers transmit data to the nano-interface and then it transmits to the nano-gateway. Finally, the data is transmitted to the base station for data processing. The transmission processes are executed over nano-links.

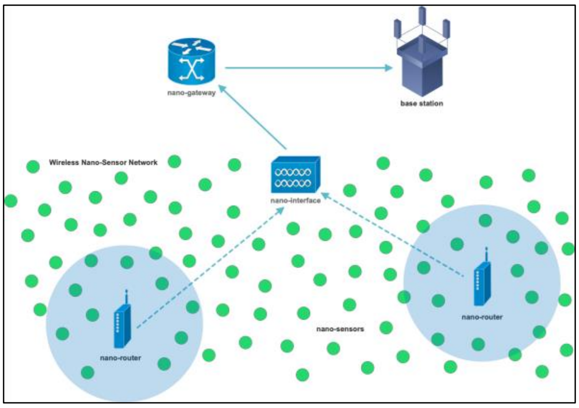

Figure 1 An illustration of a WNSN architecture

The communication range of nano-devices in a WNSN is very limited therefore, multi-hop routing protocols are used in order to provide data transmission from nano-sensors to nano-routers [6]. Although multi-hop routing is a basic routing scheme for WNSNs, the packet transmission process consumes more energy than direct transmission. On the other hand, network traffic is a critical problem to be solved in multi-hop routing. While the packets are transmitting to a nano-router over hops, the network load should be balanced for lossless communication.

According to the study of Gulec (2022) [2], a nano-sensor node consumes 0.88 nJ while receiving a packet having 22 kB size, and 8.8 nJ while transmitting a packet in a WNSN. Therefore, in multi-hop routing, it takes 9.68 nJ in total for transmitting a packet. Assuming the nano-router is 5 hops further from a nano-sensor node on average, it takes 48.4 nJ in total. This means more energy consumption on a multi-hop routing scheme despite sending a packet to a nano-router directly.

Hence, in this paper, a localization algorithm is proposed for WNSNs that finds the required nano-routers and estimates their virtual coordinates ensuring direct data transmission from a nano-sensor to these nano-routers and avoiding higher energy consumption and packet loss by using multi-hop routing.

Due to the nano-sensor nodes not having a GPS module, they can not determine their real coordinates on the network as in traditional sensor networks. According to the work of Zhou et al. (2017) [7], the nano-sensor node distances can be calculated using electromagnetic pulse durations. Pierobon et al. (2014) [8] proposed an energy harvesting routing framework for WNSNs in THz band by using the distances and the coordinates of the nano-nodes. Therefore, the location information mentioned in this study is the relative positions of the nodes to each other on the network which is specified as their virtual coordinates.

In order to estimate the numbers and their virtual coordinates of these required nano-routers, a machine learning technique – the *k*-means clustering method is used. According to the network size, the *k* parameter is predicted and the cluster heads are chosen as nano-routers. For measuring the efficiency and the accuracy of the proposed algorithm, the results are compared with the hierarchical clustering method which is another unsupervised machine learning approach. To the best of our knowledge, there exists no machine learning-supported nano-router localization algorithm for WNSNs of IoNT applications in the literature.

The main contribution of this paper is summarized as follows.

- The proposed algorithm estimates the number of required nano-routers on a WNSN and the best coordinates to locate them by machine learning method that can connect with many nano-sensors. Therefore, the nano-sensors can easily transmit the data they sense from the environment to these nano-routers directly despite transmitting it to the uttermost nano-router by multi-hop routing.
- The proposed algorithm increases the node coverage by the estimated nano-routers therefore, any nano-node can easily find the nearest nano-router.
- The proposed algorithm saves the energy of the nodes on the network which prolongs the network lifetime.
- The proposed model prevents the nano-network from problems caused by multi-hop routing.

The rest of the paper is organized as follows. Related works on node localization, machine learning models on sensor networks and WNSN applications are summarized in Section 2. The methodology and the proposed algorithm of the current study are given in Section 3. The simulation and the results are given in Section 4. Finally, Section 5 concludes the study.

2. RELATED WORKS

Node localization is one of the critical issues that should be taken into account in Wireless Sensor Networks (WSNs) related to network design and topology which causes faults, low performance, scalability, latency and coverage problems [9].

To this end, Nain et al. (2022) [10] proposed a range-based model for underwater WSNs using both particle swarm optimization and crow search optimization which reduces localization latency and provides accuracy. Yu et al. (2023) [11] proposed a quantum annealing bat algorithm to improve localization accuracy and applicability for WSNs.

Similarly, a node localization model is developed using the salp swarm algorithm for WSNs in Kanoosh et al. (2019) [12] which outperforms the other localization algorithms in terms of computing time, the number of nodes and localization error. Sekhar et al. (2021) [13] proposed a metaheuristics-based node localization technique for WSNs that aims to find the coordinates of unknown nodes by the anchor nodes with minimum error and maximum accuracy. Javed et al. (2022) [14] proposed a mobile node localization algorithm for WSNs that improves positioning accuracy. Aroba et al. (2023) [15] proposed an algorithm to address the problem of determining sensor node localization with the minimum error when the data is transmitted over the wireless channel.

In the literature, machine learning-based methods also have been developed for WSNs in several areas. Thereby, Galal and Hesselbach (2020) [4] analyzed and classified nano-network traffic using five supervised machine learning models. Dampage et al. (2022) [16] proposed a system to detect forest fires at the initial stage using WSNs and a machine learning regression model. Bacanin et al. (2022) [17] proposed a deep learning-based model to predict air quality using Dragonfly optimization for node localization in WSNs. Esmaeili et al. (2022) [18] designed a combined model for WSNs using metaheuristics and machine learning methods for clustering-based network routing.

A *k*-means clustering-based node localization algorithm for WSNs is proposed by Khediri et al. (2020) [19] where single-hop communication is used in intra-clusters and multi-hop communication is used in inter-clusters. Mahmood et al. (2022) [20] designed a fault detection and energy-efficient routing system for WSNs using routing-based reinforcement learning. Li et al. (2022) [21] proposed a model using *k*-means clustering for routing in underwater wireless sensor networks (UWSNs). Sathyamoorthy et al. (2022) [22] proposed a Q-learning-based clustering and load-balancing technique using *k*-means for WSNs that maximizes throughput, packet delivery ratio and minimizes end-to-end delay and energy consumption.

Recently, several methods have been developed for WNSNs problems in many fields. Xu et al. (2021) [23] developed a multi-hop routing protocol for intra-body WNSNs which estimates link states. Gulec and Sahin (2023) [24] proposed a Red Deer Algorithm based nano-sensor node clustering method for WNSNs. Garcia-Sanchez et al. (2023) [25] proposed a multi-hop routing scheme using reinforcement learning in body WNSNs.

3. METHODOLOGY

3.1. The Machine Learning Method

Node localization is one of the most difficult challenges for sensor networks to achieve due to the harsh or dangerous environment in which the nodes are deployed. To find the accurate location of the nodes, machine learning approaches can be helpful [26].

For this purpose, in the proposed model the prediction of the number of required nano-routers depending on WNSN size and their best coordinates is predicted with the *k*-means clustering method which is one of the most known unsupervised machine learning models that gives the center points (centroids) of the clusters [27, 28].

The *k*-means clustering model divides the points into *k* clusters therefore, it is important to find the optimal *k* number. The model finds centroids that cover the points close to them and include the points in the proper clusters. These points are located in these clusters by using Euclidean distance. This machine learning model iteratively computes the mean of the clusters until the minimum number of optimal centroids is found [26].

In this paper, *k*-means clustering as a machine learning model is used for clustering the nano-nodes in order to find optimal centroids as the nano-routers that should be placed on the network for ensuring efficient direct data transmission on WNSNs.

3.2. The Proposed Algorithm

In the proposed model, first of all, the initial nano-router (R_0) is placed in the coordinate [0, 0] on the network. In a 0.0015 meters transmission range, many of the nano-sensor nodes on the network can not communicate to this nano-router directly. Therefore, new nano-routers need to be placed at suitable coordinates on the network to ensure direct data transmission.

When the proposed algorithm is initialized, it is assumed that the nano-sensor nodes know their IDs and virtual coordinates. The nano-sensor nodes first broadcast *NEIGHBOUR* messages to know their neighbour nodes. When a nano-sensor node receives this message, adds the sender node to its neighbourhood list (Γ_m). This operation is repeated several times until all nano-sensor nodes know their neighbours. Finally, the nano-sensors send *NEIGHBOUR_DONE* message to the initial nano-router.

Then the initial nano-router (R_0) broadcasts *CALCULATE* message including its coordinates to collect the distances of the nano-sensor nodes. When a nano-sensor node receives this message, calculates its distance by the Euclidean distance given in Equation 1 where x_m is its x-coordinate, y_m is its y-coordinate, x_r is the nano-router's x-coordinate and y_r is the nano-router's y-coordinate. After calculating this value, it sends back to R_0.

$$d = \sqrt{(x_m - x_r)^2 + (y_m - y_r)^2} \qquad (1)$$

The initial nano-router receives all the distance values from all nano-sensor nodes on the network and then sends these values to the nano-gateway (*NG*). The nano-gateway calculates the *k* value for the *k*-means clustering method. In this step, *NG* finds the *k* value by the elbow technique of the *k*-means clustering method. Then it sends the cluster information to R_0 by *CLUSTER* message.

As an example, the graph of an elbow technique on a sample WNSN having 500 nodes for finding the optimal cluster number on the network is given in Figure 2. According to the figure, the optimal cluster number is calculated as 4.

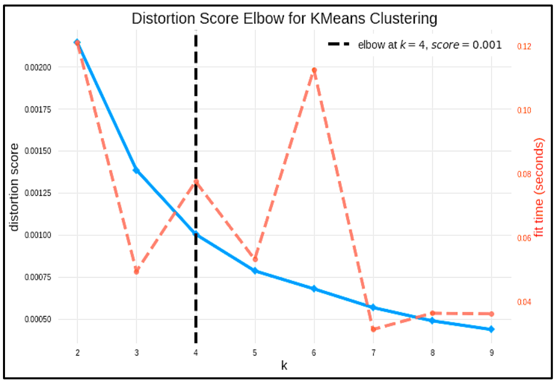

Figure 2 The elbow technique for calculating optimal cluster number on a sample WNSN

In the first iteration of placing new nano-routers on the network, the proposed algorithm finds the optimal coordinates on each cluster and directs new nano-routers to these coordinates. Unfortunately, the nano-routers have a 0.002 meters transmission range. Therefore, in this step, the node coverage may not be ensured yet. In other words, these new nano-routers may not cover all of the nano-sensor nodes.

Hence, the proposed algorithm checks if there exists any nano-sensor node that has no direct connection to any nano-router. Thus, the nano-routers broadcast *COVERAGE* messages including their information on the medium. A nano-sensor node that receives this message, adds this nano-router to its router list (Γ_{mr}) and sends *COVERED* message including its information to the sender nano-router (R_i). If a nano-node is covered by several clusters, it adds all nano-routers having a direct connection to its router list (Γ_{mr}).

If there exists any nano-sensor node that is not covered yet by *checkCoverage()* function, the nano-gateway executes another *k*-means clustering method for a cluster by calculating the distances between the nano-sensor nodes and the relevant cluster heads to place more nano-routers instead of the previous ones and finds their best coordinates to ensure more node coverage.
In the last step of the algorithm, all of the new nano-routers are placed on the network that they cover almost all nano-sensor nodes. If the node coverage conditions are satisfied, then the algorithm terminates. The first part of the proposed algorithm which is executed by nano-sensors is given in Algorithm 1, the second part which is executed by nano-routers is given in Algorithm 2 and the last part of the proposed algorithm executed by the nano-gateway is given in Algorithm 3, respectively.

Algorithm 1 Nano-sensor part

data:
  $\Gamma_m \leftarrow$ neighbour list of nano-sensor node m
  $\Gamma_{mr} \leftarrow$ nano-router list of node m
  $R_0 \leftarrow$ initial nano-router placed at [0, 0]
  $d_m \leftarrow$ the distance between node m and R_0
initially:
  $\Gamma_m \leftarrow \emptyset,\ \Gamma_{mr} \leftarrow \emptyset$
upon the algorithm started:
  broadcast *NEIGHBOUR* msg to Γ_m
upon receiving *NEIGHBOUR* msg from node *n*:
  add node n to Γ_m
  send *NEIGHBOUR_DONE* msg to R_0
upon receiving *CALCULATE* msg from R_0 or node *n*:
  calculate d_m
  send d_m to R_0
upon receiving *COVERAGE* msg from R_i:
  add R_i to Γ_{mr}
  send *COVERED* msg to R_i

Algorithm 2 Nano-router part

data:
  $NG \leftarrow$ nano-gateway
upon the algorithm started:
  wait until nano-sensors finish the first process
upon receiving *NEIGHBOUR_DONE* msg from node *n*:
  broadcast *CALCULATE* msg
upon receiving d_n values from $\forall n$:
  send $\forall d_n$ values to NG
upon receiving *CLUSTER* msg from NG:
  send $[x_i, y_i] \in R_i$ to n_i by *COVERAGE* msg

Algorithm 3 Nano-gateway part

data:
  finished $\leftarrow$ *true* if the algorithm terminates
initially:
  finished $\leftarrow$ *false*
upon receiving d_n values from R_0:
  estimate the number of required R_i
  cluster nodes by d_n using *k*-means
  send $[x_i, y_i] \in R_i$ to R_0 by *CLUSTER* msg
if checkCoverage() == FALSE:
  repeat clustering nodes by d_u using *k*-means
  send $[x_j, y_j] \in R_j$ to R_0 by *CLUSTER* msg
else:
  finished $\leftarrow$ *true*

4. SIMULATIONS AND RESULTS

In order to estimate the number of required nano-routers and their virtual coordinates on the network, NS-3 [29] simulation tool and Nano-Sim [30] framework are both used for the simulations. In NS-3 simulator, basically, a nano-network is formed by the nano-nodes, the nano-routers, nano-interfaces and the nano-gateway. To illustrate all of the nano-devices and the network, Python packages, Networkx [31] and Matplotlib [32] are also used in the simulations.

Besides, different network topologies have been generated in different network sizes for the simulations. Therefore, the proposed algorithm was conducted 100 times on these topologies having 500, 750, 1000 and 2000 nano-nodes. The size of the network area is 0.01×0.01 meters, the transmission range of nano-sensors is 0.0015 meters and the transmission range of nano-routers is 0.002 meters. All of the simulation parameters are given in Table 1.

Table 1 Simulation parameters

Simulator	NS-3, Nano-Sim
Python Package	Networkx, Matplotlib
Network Area	0.01 x 0.01 m^2
Transmission Range of Nano-Nodes	0.0015 m
Transmission Range of Nano-Routers	0.002 m
Packet Size	22 kB
Topologies	100 on each simulation
Number of Nano-Sensor Nodes	500, 750, 1000, 2000

Table 1 Simulation parameters (continue)

A sample nano-network having 250 nodes is illustrated in Figure 3. According to the figure, the rest of the nano-sensor nodes are given that are not in the transmission range of the initial nano-router which is placed in [0, 0]. These nano-sensor nodes are clustered according to the distances to the initial nano-router by the *k*-means clustering method. In the figure, for this sample network, there are 3 clusters having nano-sensor nodes that are colored red, green and blue.

After clustering these nano-sensor nodes, the optimal centroids/cluster heads are placed in each cluster as nano-routers which are colored orange. These nano-routers have 0.002 meters transmission range, hence, they do not cover all of the nano-sensor nodes in their own clusters. According to Figure 3, it is clearly seen that placing a nano-router in each cluster is not enough to ensure node coverage. Therefore, the *k*-means clustering method is executed once more on each cluster for placing new nano-routers instead of the centroid cluster heads due to the transmission range of nano-routers.

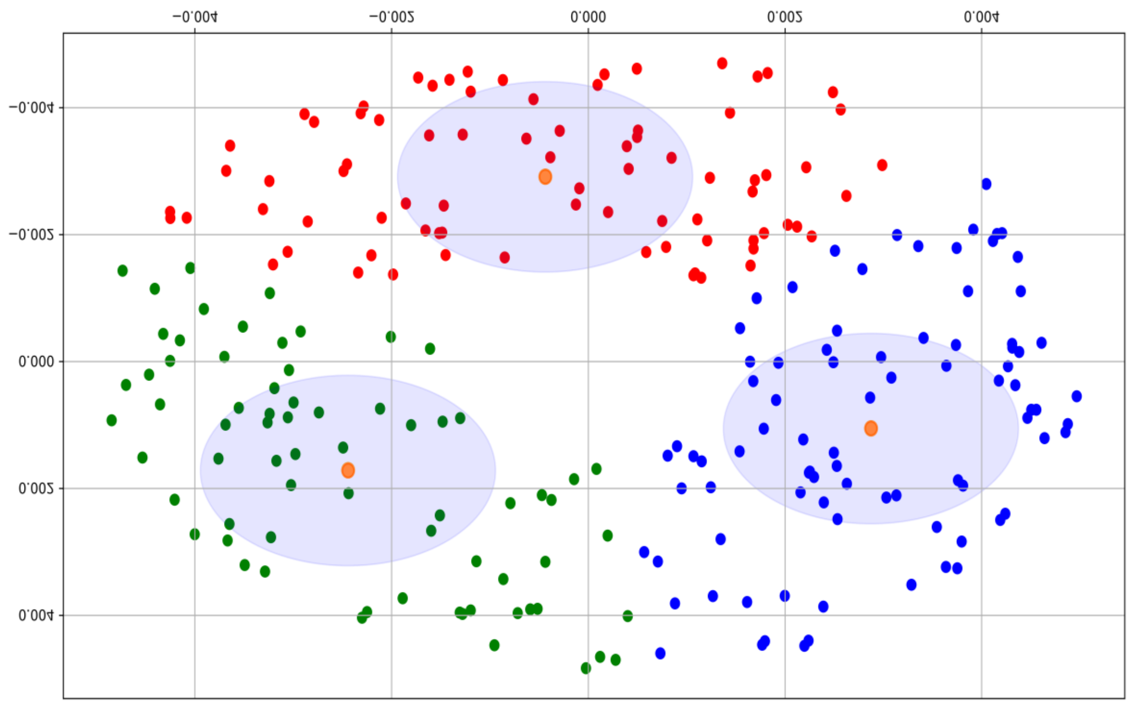

Figure 3 The first iteration of nano-router localization by the proposed algorithm on a sample WNSN

In Figure 4 and Figure 5, it is shown that there are more nano-routers that are placed on the network for each cluster. In Figure 6, the final network topology of a sample WNSN having 250 nano-nodes and 7 nano-routers is illustrated. Similarly, another WNSN having 2000 nano-nodes and 17 nano-routers that are placed by the proposed algorithm is given in Figure 7.

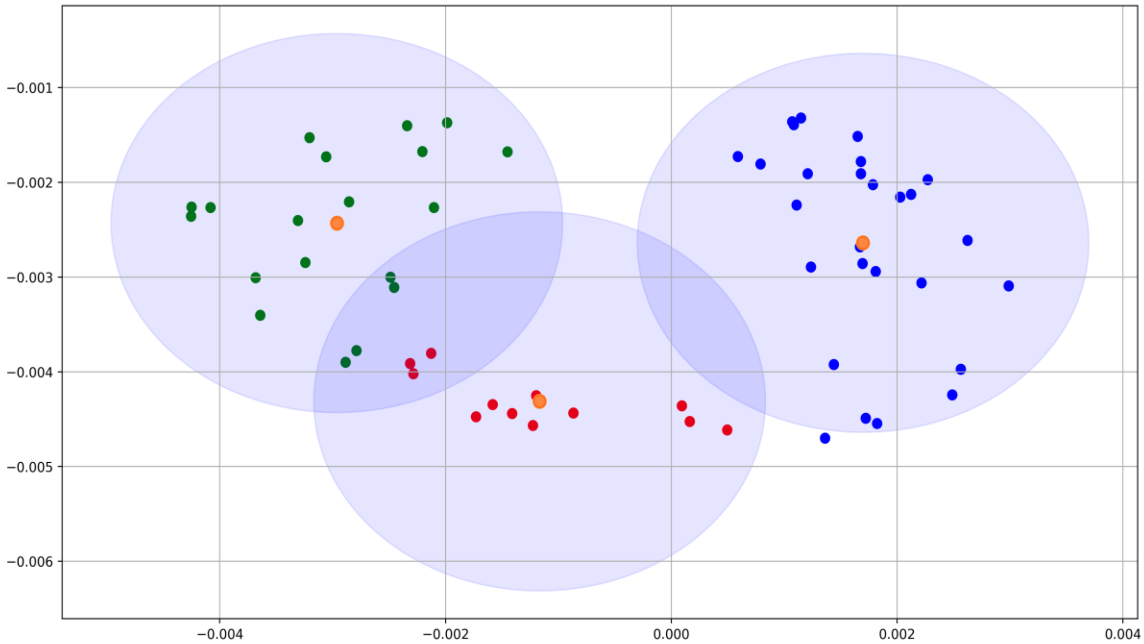

Figure 4 The second iteration of nano-router localization algorithm for red-colored clusters

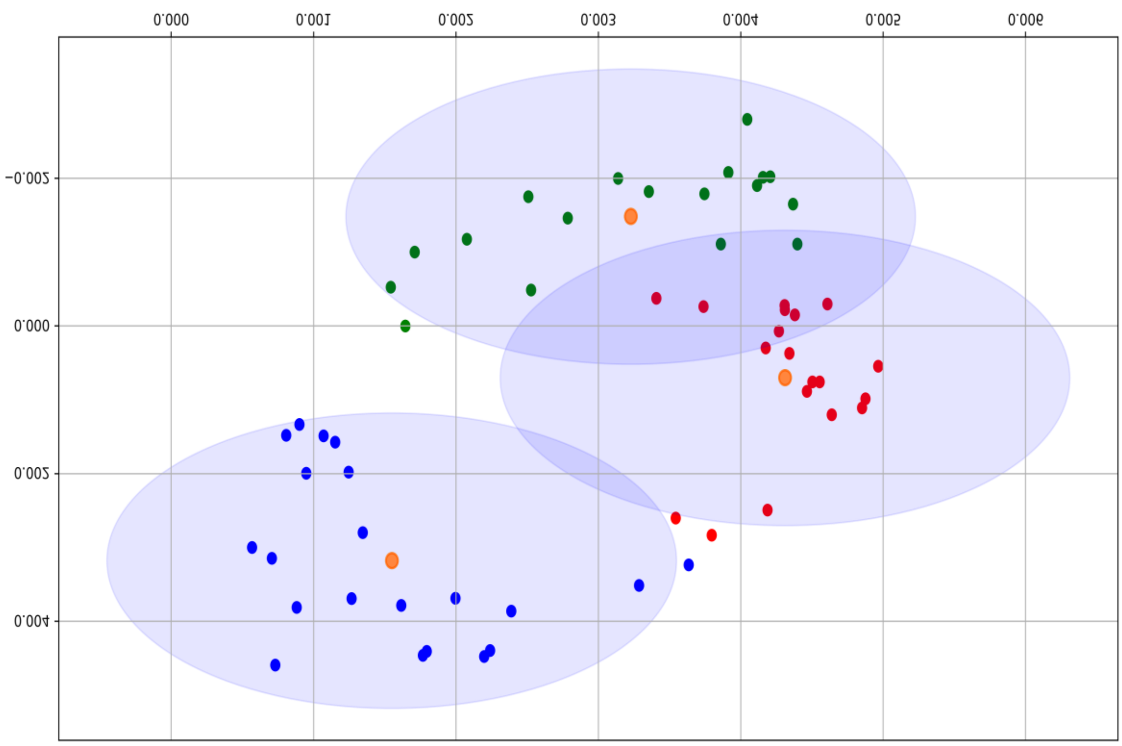

Figure 5 The second iteration of nano-router localization algorithm for blue-colored clusters

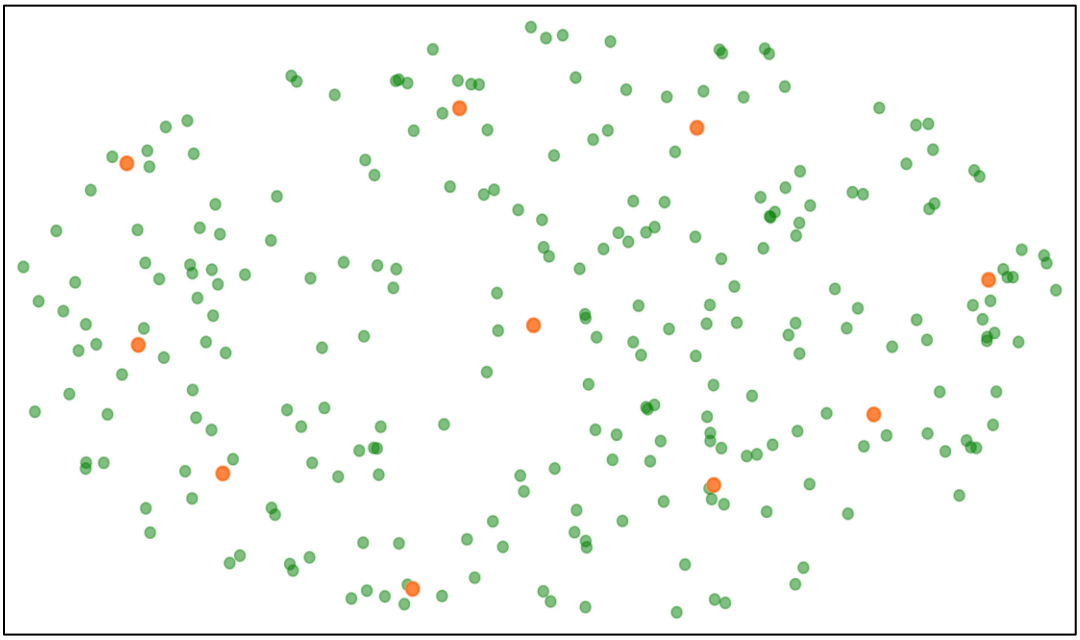

Figure 6 The final nano-router localization using the proposed algorithm on a sample WNSN

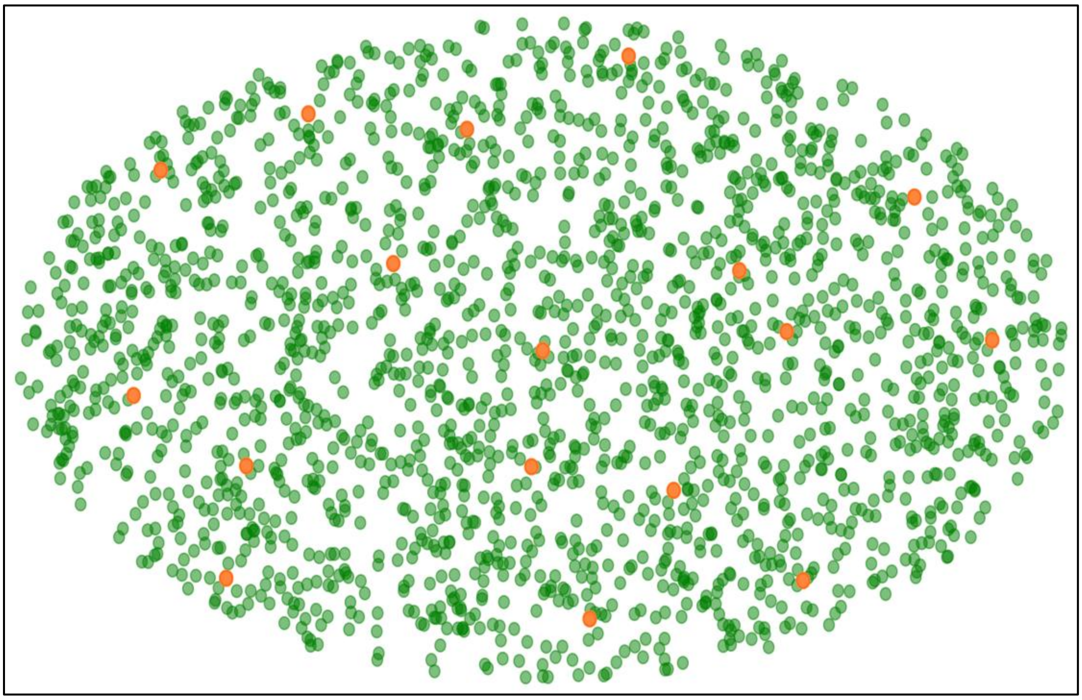

Figure 7 The nano-routers placed by the proposed algorithm on another WNSN

According to the average simulation results of the proposed algorithm given in Table 2, when a nano-network has 500 nodes, the nano-router localization takes 0.096 seconds while 461.43 nano-nodes are covered having 92.28% coverage percentage on average. When a nano-network has 750 nano-sensor nodes, the nano-router localization process takes 0.103 seconds while 695.2 nodes are covered having 92.66% percentage of coverage on average. If the WNSN has 1000 nano-nodes, the time spent for nano-router localization is 0.13 seconds, 980.31 nodes are covered while the node coverage is 98.03% on average. If a WNSN consists of 2000 nano-nodes, it takes 0.175 seconds to find new nano-routers while 1953.41 nodes are covered as 97.67% percentage on average.

Table 2 Average simulation results

Node Count	Total Time (s)	Covered Node Count	Coverage (%)
500	0.096	461.43	92.28
750	0.103	695.2	92.66
1000	0.13	980.31	98.03
2000	0.175	1953.41	97.67

According to the average simulation results of the hierarchical clustering method given in Table 3, when a nano-network has 500 nodes, the nano-router localization takes 0.113 seconds while 394.29 nano-nodes are covered having 78.85% coverage percentage on average. When a nano-network has 750 nano-sensor nodes, the nano-router localization process takes 0.124 seconds while 587.08 nodes are covered having 78.27% percentage of coverage on average. If the WNSN has 1000 nano-nodes, the time spent for nano-router localization is 0.157 seconds, 818.84 nodes are covered while the node coverage is 81.88% on average. If a WNSN consists of 2000 nano-nodes, it takes 0.224 seconds to find new nano-routers while 1640.29 nodes are covered as 82.01% percentage on average.

Table 3 Average hierarchical clustering results

Node Count	Total Time (s)	Covered Node Count	Coverage (%)
500	0.113	394.29	78.85
750	0.124	587.08	78.27
1000	0.157	818.84	81.88
2000	0.224	1640.29	82.01

In brief, the proposed algorithm achieves 13.43% more coverage over 500 nodes, 14.39% more coverage over 750 nodes, 16.15% more coverage over 1000 nodes and 15.66% more coverage over 2000 nodes on average by placing the proper nano-routers on WNSNs according to the comparison results given in Figure 8.

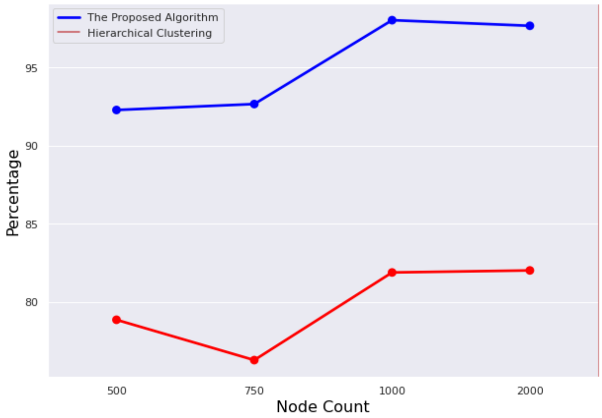

Figure 8 The coverage percentage results of the proposed algorithm and hierarchical clustering

5. CONCLUSION

In this paper, a machine learning-supported nano-router localization algorithm is proposed for providing direct communication between nano-sensor nodes and nano-routers on WNSNs running on IoNT applications. For finding the optimal nano-routers, *k*-means clustering is used as a machine learning method iteratively for providing maximum node coverage.

The proposed algorithm has been developed and tested in NS-3 simulator and Nano-Sim framework under different network topologies having different network sizes. Each simulation is conducted 100 times with different simulation setups for obtaining average results. For the illustration of the networks, Python packages, Networkx and Matplotlib are also used. The obtained simulation results of the proposed algorithm have been compared with the hierarchical clustering method which is another unsupervised machine learning approach.

According to the simulation results, the proposed algorithm successfully predicts the required number of nano-routers and estimates their optimal virtual coordinates that ensure higher node coverage on a WNSN for providing direct communication between nano-sensor nodes and the nano-routers. The proposed algorithm increases node coverage up to 98.03% by placing new nano-routers on the network. Besides, the proposed algorithm avoids higher energy consumption of the nano-sensor nodes for transmitting the data packets on multi-hop routing to the available nano-router.

Node localization is a critical problem to be considered in all types of sensor networks. For a better solution to the localization problem in WNSNs, mobile nano-routers may be used for redirecting the estimated coordinates by the nano-gateways for future studies.

Energy-efficient multi-hop routing algorithms may be used on WNSNs in cases where there are not enough nano-routers can be placed on the network. Therefore, both nano-router localization and multi-hop routing hybrid algorithms will be developed for future studies.

Funding
The author has no received any financial support for the research, authorship or publication of this study.

Authors' Contribution
O.G: Conceptualization, literature review, methodology/study design, software development, simulation, analysis, validation, writing – original draft, writing – review and editing, visualization, supervision.

The Declaration of Conflict of Interest/ Common Interest
No conflict of interest or common interest has been declared by the authors.

The Declaration of Ethics Committee Approval

This study does not require ethics committee permission or any special permission.

The Declaration of Research and Publication Ethics

The authors of the paper declare that they comply with the scientific, ethical and quotation rules of SAUJS in all processes of the paper and that they do not make any falsification on the data collected. In addition, they declare that Sakarya University Journal of Science and its editorial board have no responsibility for any ethical violations that may be encountered, and that this study has not been evaluated in any academic publication environment other than Sakarya University Journal of Science.

REFERENCES

[1] A. O. Balghusoon, M. Saoucene, “Routing protocols for wireless nanosensor networks and internet of nano things: A comprehensive survey”,IEEE Access, 8, 200724-200748, 2020.

[2] O. Gulec, “Extending lifetime of Wireless Nano-Sensor Networks: An energy efficient distributed routing algorithm for Internet of Nano-Things”, Future Generation Computer Systems, 135, 382-393, 2022.

[3] A. Rizwan, A. Zoha, R. Zhang, W. Ahmad, K. Arshad, N. A. Ali, Q. H. Abbasi, “A review on the role of nano-communication in future healthcare systems: A big data analytics perspective”, IEEE Access, 6, 41903-41920, 2018.

[4] A. Galal, X. Hesselbach, “Machine Learning Models for Traffic Classification in Electromagnetic Nano-Networks”, IEEE Access, 10, 38089-38103, 2022.

[5] M. A. Akkaş, R. Sokullu, “Wireless Communications Beyond 5 g: Teraherzwaves, Nano-Communications and the Internet of Bio-Nano-Things”, Wireless Personal Communications, 126, 3543–3568, 2022.

[6] A. Galal, X. Hesselbach, “Probability-based path discovery protocol for electromagnetic nano-networks”, Computer Networks, 174, 107246, 2020.

[7] L. Zhou, G. Han, L. Liu, “Pulse-based distance accumulation localization algorithm for wireless nanosensor networks”, IEEE Access, 5, 14380-14390, 2017.

[8] M. Pierobon, J. M. Jornet, N. Akkari, S. Almasri, I. F. Akyildiz, “A routing framework for energy harvesting wireless nanosensor networks in the Terahertz Band”, Wireless Networks, 20, 1169-1183, 2014.

[9] P. Yadav, S. C. Sharma, “A Systematic Review of Localization in WSN: Machine Learning and Optimization-Based approaches”, International Journal of Communication Systems, e5397, 2022.

[10] M. Nain, N. Goyal, L. K. Awasthi, A. Malik, “A range based node localization scheme with hybrid optimization for underwater wireless sensor network”, International Journal of Communication Systems, 35(10), e5147, 2022.

[11] S. Yu, J. Zhu, C. Lv, “A Quantum Annealing Bat Algorithm for Node Localization in Wireless Sensor Networks”, Sensors, 23 (2), 782, 2023.

[12] H. M. Kanoosh, E. H. Houssein, M. M. Selim, “Salp Swarm Algorithm for Node Localization in Wireless Sensor Networks”, Journal of Computer

Networks and Communications, vol. 2019, 1028723, 2019.

[13] P. Sekhar, E. L. Lydia, M. Elhoseny, M. Al-Akaidi, M. M. Selim, K. Shankar, “An effective metaheuristic based node localization technique for wireless sensor networks enabled indoor communication”, Physical Communication, 48, 101411, 2021.

[14] I. Javed, X. Tang, M. A. Saleem, M. U. Sarwar, M. Tariq, C. S. Shivachi, “3D localization for mobile node in wireless sensor network”, Wireless Communications and Mobile Computing, 2022)

[15] O. J. Aroba, N. Naicker, T. T. Adeliyi, “Node Localization in Wireless Sensor Networks using a Hyper-Heuristic DEEC-Gaussian Gradient Distance Algorithm”, Scientific African, e01560, 2023.

[16] U. Dampage, L. Bandaranayake, R. Wanasinghe, K. Kottahachchi, B. Jayasanka, “Forest fire detection system using wireless sensor networks and machine learning”, Scientific Reports, 12 (1), 46, 2022.

[17] N. Bacanin, M. Sarac, N. Budimirovic, M. Zivkovic, A. A. Al Zubi, A. K. Bashir, “Smart wireless health care system using graph LSTM pollution prediction and dragonfly node localization”, Sustainable Computing: Informatics and Systems, 35, 100711, 2022.

[18] H. Esmaeili, B. M. Bidgoli, V. Hakami, “CMML: Combined metaheuristic-machine learning for adaptable routing in clustered wireless sensor networks”, Applied Soft Computing, 118, 108477, 2022.

[19] S. El Khediri, W. Fakhet, T. Moulahi, R. Khan, A. Thaljaoui, A. Kachouri, “Improved node localization using K-means clustering for Wireless Sensor Networks”, Computer Science Review, 37, 100284, 2020.

[20] T. Mahmood, J. Li, Y. Pei, F. Akhtar, S. A. Butt, A. Ditta, S. Qureshi, “An intelligent fault detection approach based on reinforcement learning system in wireless sensor network”, The Journal of Supercomputing, 78(3), 3646-3675, 2022.

[21] L. Li, Y. Qiu, J. Xu, “A K-means clustered routing algorithm with location and energy awareness for underwater wireless sensor networks”, Photonics, Vol. 9, No. 5, p. 282, MDPI, 2022.

[22] M. Sathyamoorthy, S. Kuppusamy, R. K. Dhanaraj, V. Ravi, “Improved K-means based q learning algorithm for optimal clustering and node balancing in WSN”, Wireless Personal Communications, 122(3), 2745-2766, 2022.

[23] J. Xu, Y. Zhang, J. Jiang, J. Kan, “A multi-hop routing protocol based on link state prediction for intra-body Wireless Nanosensor Networks”, Ad Hoc Networks, 116, 102470, 2021.

[24] O. Gulec, E. Sahin, “ Red Deer Algorithm based nano-sensor node clustering for IoNT”, Journal of Network and Computer Applications, 103591, 2023.

[25] A. J. Garcia-Sanchez, R. Asorey-Cacheda, J. Garcia-Haro, J. L. Gomez-Tornero, “Dynamic Multihop Routing in Terahertz Flow-Guided Nanosensor Networks: A Reinforcement Learning Approach”, IEEE Sensors Journal, vol. 23, no. 4, pp. 3408-3422, 2023.

[26] P. Nayak, K. G. Swetha, S. Gupta, K. Madhavi, “Routing in wireless sensor

networks using machine learning techniques: Challenges and opportunities", Measurement, 178, 2021.

[27] D. Stiawan, M. E. Suryani, M. Y. Idris, M. N. Aldalaien, N. Alsharif, R. Budiarto, "Ping flood attack pattern recognition using a K-means algorithm in an Internet of Things (IoT) network", IEEE Access, 9, 2021.

[28] H. Mahboubi, B. Stéphane, A. G. Aghdam, "A machine learning assisted method for coverage optimization in a network of mobile sensors", IEEE Transactions on Industrial Informatics, 2022.

[29] NS-3. Discrete-event network simulator for Internet systems [Online] Available: https://www.nsnam.org

[30] Nano-Sim. [Online] Available: https://telematics.poliba.it

[31] Networkx. Network analysis in Python [Online] Available: https://networkx.org

[32] Matplotlib. Visualization with Python [Online] Available: https://matplotlib.org