International Journal of Web of Multidisciplinary Studies
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
website: https://ijwos.com
Volume 3 Issue 4 , April 2026
OPEN ACCESS
E-ISSN : 3049-2424
DOI: https://doi.org/10.71366/ijwos

TINYML ENABLED EDGE AI BASED ENERGY EFFICIENT WSN USING DIGITAL TWIN SIMULATION

Anitha G[1], Dr.R.Sittalatchoumy[2]
[1] *Communication system, College of Engineering, Guindy,India*
[2] *Associate Professor, College of Engineering, Guindy,India*

Article Info

Article History:

Published:27 April 2026

Publication Issue:
Volume 3, Issue 4
April-2026

Page Number:
236-245

Corresponding Author:
Anitha G

Abstract:
In present time WSNs are very much used for continuous health care but traditional WSN systems do have high energy issue which they face because of continuous data transfer. In this paper we present an energy efficient patient monitoring system that we have put together using Tiny Machine Learning (TinyML) and the Digital Twin concept. We use physiological parameters like temperature, heart rate, and blood oxygen saturation (SpO2) to monitor patient health. Also we have put in a TinyML based edge decision which classifies the data into normal and abnormal right at the sensor node. As opposed to what we see in present WSNs which send out all the collected data, our put forth approach only sends out abnormal data which in turn greatly reduces unneeded communication. Also we have developed a Digital Twin model in MATLAB which we use to do real time patient monitoring and study system behavior using data driven input. We evaluate the performance of the put forth system by comparing it with a traditional WSN model in terms of data transfer and energy use. We present that the TinyML based WSN does in fact reduce the transmission rate and achieve great energy savings while at the same time does not sacrifice monitoring quality. This approach we put forth improves the efficiency and scalability of health care monitoring systems which in turn makes it very suitable for real time and resource constrained environment.

Keywords: WSN, TinyML, Digital Twin, Edge Computing, Patient Monitoring, Energy Efficiency

1.Introduction

For continuous and remote patient monitoring, Wireless Sensor Networks (WSNs) have become an important technology in modern healthcare systems. These systems use several sensor nodes to gather physiological data like heart rate, body temperature, and blood oxygen saturation (SpO_2), which lets doctors see how a patient's health is changing in real time. But traditional WSN-based monitoring systems have a lot of problems, especially when it comes to using a lot of energy because sensor nodes keep sending data to centralized servers. This shortens the life of the network and makes it less useful in places where resources are limited.Recent progress in edge computing and Tiny Machine Learning (TinyML) has made it possible for sensor nodes to process data intelligently right at the node level. TinyML lets small machine learning models run on low-power embedded systems.

Recent developments in edge computing and Tiny Machine Learning (TinyML) have made it possible to process data intelligently right at the sensor node level. This is a way to get around these problems. TinyML lets small machine learning models run on low-power embedded devices, so decisions can be made locally without having to constantly talk to the cloud. When sensor data is analyzed at the edge, only important or unusual information can be sent, which cuts down on communication and energy use by a lot.

Also, the idea of a "digital twin" has gotten a lot of attention as a great way to model and study real-world systems in a virtual space. A Digital Twin makes a virtual copy of the physical system that is always up-to-date. This lets you monitor, analyze, and improve the system without having to touch the hardware directly. Researchers can use this to efficiently simulate patient conditions and test how well the system works in healthcare settings.

This paper proposes a TinyML-based energy-efficient wireless sensor network (WSN) for monitoring patient health that works with a Digital Twin simulation model. At the edge node, the system sorts physiological data into normal and abnormal conditions in real time. Using dataset-driven simulation to mimic real-world behavior, a Digital Twin based on MATLAB is made. We compare the proposed method to a standard WSN model to see if it improves data transmission speed and energy use.

The results show that combining TinyML and Digital Twin greatly improves system performance by cutting down on unnecessary data transmission and making the system more energy efficient. This makes it a good choice for smart healthcare applications of the future.

2 . Objective

The main goal of this project is to create an energy-efficient patient monitoring system by combining Digital Twin technology, Wireless Sensor Networks (WSNs), and Tiny Machine Learning (TinyML). In traditional healthcare monitoring systems, sensor nodes constantly send physiological data like temperature, heart rate, and blood oxygen saturation (SpO_2) to a central server. This uses a lot of energy and shortens the life of the network. To fix this problem, the proposed system adds a WSN-based framework that collects real-time physiological parameters and processes them at the sensor node. The system uses TinyML-based edge intelligence to classify patient conditions into normal and abnormal states on the device. This means that data doesn't have to be sent all the time. This selective communication strategy makes sure that only important or unusual data is sent, which cuts down on communication overhead and makes the system use less energy.The work also aims to create a Digital Twin model in MATLAB that can mimic how the real patient monitoring system works in a virtual setting. The Digital Twin uses inputs from datasets to mimic how sensors work in real time. This makes it possible to monitor, visualize, and analyze performance without having to build physical hardware. A comparative analysis is performed between traditional WSN and TinyML-based WSN to assess enhancements in transmission frequency and energy efficiency. The results show that the suggested method saves a lot of energy while still allowing for reliable health monitoring. This work aims to show that combining TinyML and Digital Twin technologies makes systems more efficient, scalable, and reliable, which makes them perfect for next-generation smart healthcare applications.

3. Literature Review

Wireless Sensor Networks (WSNs) have been thoroughly investigated for healthcare monitoring applications because they can continuously and remotely track a patient's health. Early research concentrated on the implementation of sensor nodes for the acquisition of physiological data, including temperature, heart rate, and blood oxygen saturation, which are subsequently relayed to a centralized system for analysis. But these traditional methods depend on constant data transmission, which uses a lot of energy and shortens the life of the network.

Researchers have come up with energy-saving methods like data aggregation, duty cycling, and adaptive transmission strategies to deal with these problems. These methods do cut down on communication overhead to some extent, but they don't use smart decision-making at the sensor node level. Because of this, there is still a lot of unnecessary data transmission in many real-world situations.

Tiny Machine Learning (TinyML) has become a promising way to make devices with limited resources smart as edge computing has gotten better. With TinyML, lightweight machine learning models can be put on embedded systems, which lets data be processed and decisions be made in real time at the edge. Several studies have shown that TinyML works well for finding anomalies and keeping an eye on health. In these cases, only important or unusual data is sent to cut down on communication load and save energy.

The idea of Digital Twin has become really popular lately for modeling and simulating systems in a virtual environment. A Digital Twin is like a copy of a thing and it shows what is happening with that thing in real time. This allows us to watch what is going on analyze it and make it better without having to touch the hardware. In healthcare people have used Digital Twin models to simulate what is happening with patients predict how systems will behave and see how they will perform in situations.

With all the progress that has been made putting TinyML and Digital Twin together for healthcare monitoring that is good on energy and uses WSN is not something that many people have looked at. Most of the work that has been done. Focuses on making edge processing smarter or simulating systems but it does not combine these two things into one framework.

The Digital Twin is used in this work to fill in the gaps by suggesting a system that uses TinyML and WSN together. This system can find anomalies at the edge level. Uses a simulation that is based on MATLAB to show what patient monitoring is like in real time. Also we compare the way of doing WSN to the new way that uses TinyML to see if it is better, at sending data and using energy.

4. Proposed system

The proposed system presents an energy-efficient patient monitoring setup. It combines Wireless Sensor Networks (WSNs), Tiny Machine Learning (TinyML), and Digital Twin technology to improve modern healthcare systems. The process starts with sensor nodes. These nodes continuously collect real-time physiological data from patients, such as temperature, heart rate, blood oxygen saturation (SpO_2), and motion. The sensor nodes act as the main data collection units and form the basis of the monitoring system by generating continuous streams of health-related data.

Instead of sending this raw data directly to a central server, the proposed system includes a TinyML-based Edge AI module. This module acts as the main intelligence of the setup. It runs a lightweight machine learning model within the sensor node, allowing for local data processing and real-time decision-making. The TinyML model reviews the incoming sensor data and sorts the patient's condition into either normal or abnormal based on set thresholds or learned patterns. This edge-level intelligence greatly cuts down on the need for cloud-based processing, reduces latency, and improves the system's responsiveness.

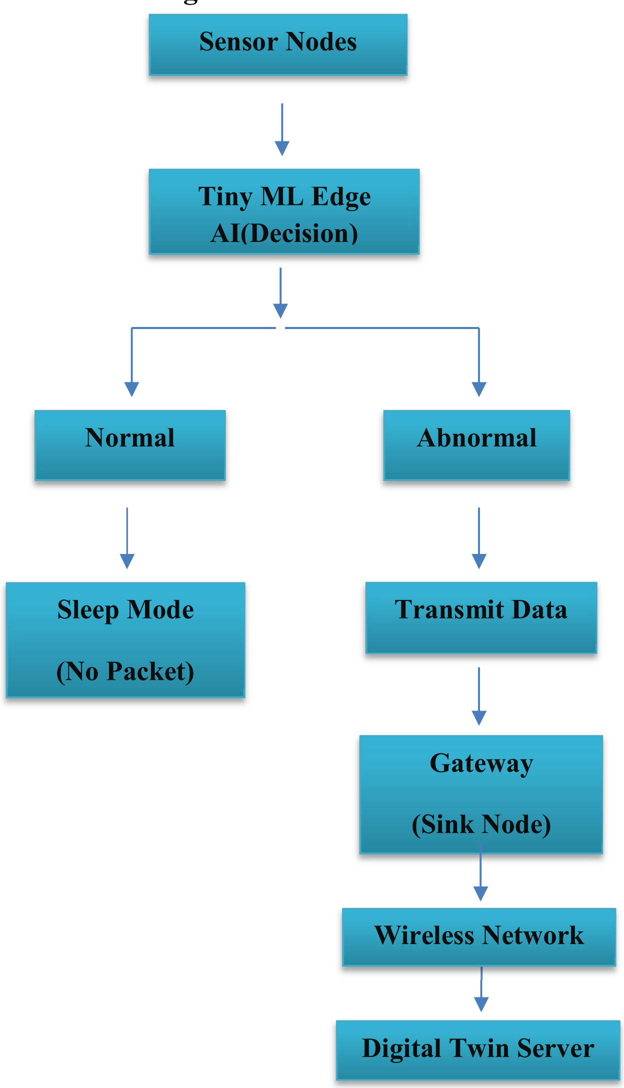
According to the classification results, the operation of the system has two paths. When the condition of the patient is normal, The sensor node does not transmit data, but rather enters low power or sleep mode thus minimizing the amount of communication (the primary cause of energy consumption in WSNs), conserving battery power and prolonging the life of the network. When an abnormal condition occurs, however; the system automatically activates data transmission. The detecting of abnormal data causes the immediate sending of the abnormal data; this will allow for timely alerts so that medical assistance can be obtained quickly as well as protecting the reliability and safety of the system for monitoring purposes.

Once the data has been transmitted, it is received at a gateway (also known as a sink node) which provides a link between the sensor network and central server. The gateway collects data from several sensor nodes and processes it in a basic way if required before sending that data on to the destination using a wireless communication network e.g. Wi-Fi or Zigbee or LoRa etc. This allows for fast and accurate transfer of data between nodes on the network.

Once all the required data has arrived at the Digital Twin server, which acts as a virtual mirror of the real patient monitoring system, the data is processed in the Digital Twin environment using the MATLAB development environment. The Digital Twin system imports all the dataset library input data (dataset-derived inputs), simulates the real-time behavior of the system using the imported dataset-derived inputs, and then replicates sensor functioning, decision-making processes, and communication methods. As a result, the Digital Twin provides an ability to graphically visualize physiological data and to evaluate the performance of the system. Most importantly, the Digital Twin system provides a flexible test and validation environment for evaluating and optimizing the entire patient monitoring system without needing to deploy any physical hardware.

The Proposed architecture shows that putting together TinyML-based edge intelligence, selective communication using Digital Twin simulation into a highly effective healthcare monitoring system that can work at scale. The alternate approach enables reducing unnecessary data transmission but still can detect abnormal conditions accurately; hence, it provides greater energy efficiency, more reliable operation, and is ideal for next-generation smart healthcare applications and_healthcare technologies.

4.1.Block Diagram



5. Results and discussion

The performance of the suggested patient monitoring system using TinyML along with WSNs is simulated via Google colab while the analysis of performance using digital twin is done with help of MATLAB simulations. The performances are then analyzed based on three factors; sensors monitoring behavior, data transfer effectiveness and power consumption respectively.

Normal data range

Sensors	Normal ranges	Abnormal conditions
Body Temperature	36 – 37.5 °C	> 38 °C (fever)
Heart Rate	60 – 100 bpm	< 50 or > 120 bpm
SpO_2 (Oxygen level)	95 – 100 %	< 92 %
Motion/ Activity	Normal body movement	Sudden fall / no movement

Table 5.1 Data range

5.1 TinyML result

Datasheet: "patient_monitoring_dataset_200_samples.csv"

Using Google Colab

1.Dataset generation using Python

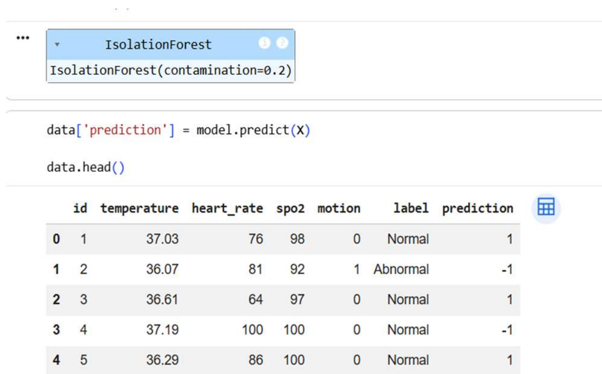
For this study, a set of data is produced through the use of Python programming language by modeling patients' physiological data such as their temperature, heartbeat, SpO2, and movement. The random data are created based on normal and abnormal values that represent various illnesses and are subjected to logical conditions.

	id	temperature	heart_rate	spo2	motion	label
0	1	37.03	76	98	0	Normal
1	2	36.07	81	92	1	Abnormal
2	3	36.61	64	97	0	Normal
3	4	37.19	100	100	0	Normal
4	5	36.29	86	100	0	Normal

2.TinyML model: Isolation Forest

The suggested framework employs Isolation Forest technique as a TinyML model for detecting anomalies in physiological sensors. It is an unsupervised learning technique that can detect anomalies without using any labels. The parameters considered by the model include temperature, heartbeat rate, SpO_2 levels, and motion activity.

An anomaly score is assigned to each input based on which the data is categorized into normal and anomaly categories. Classification is performed based on the threshold value. Thus, it becomes possible to perform decision-making at edge level with minimal computing resources.



```
IsolationForest
IsolationForest(contamination=0.2)

data['prediction'] = model.predict(X)

data.head()
```

	id	temperature	heart_rate	spo2	motion	label	prediction
0	1	37.03	76	98	0	Normal	1
1	2	36.07	81	92	1	Abnormal	-1
2	3	36.61	64	97	0	Normal	1
3	4	37.19	100	100	0	Normal	-1
4	5	36.29	86	100	0	Normal	1

Figure 5.1.1 Algorithm

3.Sensor data classification

Classification of sensor data is done in the edge node by applying the TinyML algorithm that will help classify the health status of the patient. The data collected concerning the physiological parameters such as temperature, heart rate, SpO2, and motion will be processed in real time. Depending on the results obtained from the algorithm application, the data is then sorted into two groups that include either normal or abnormal.

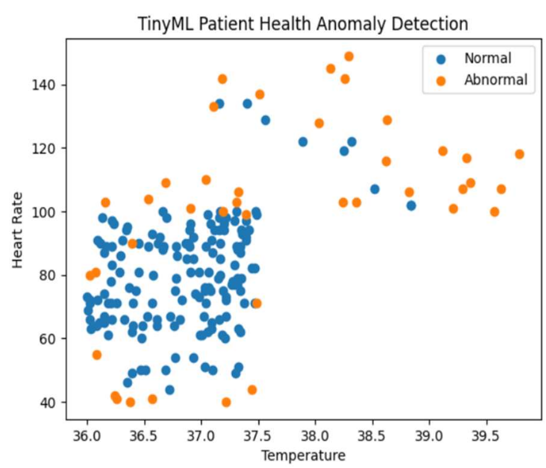
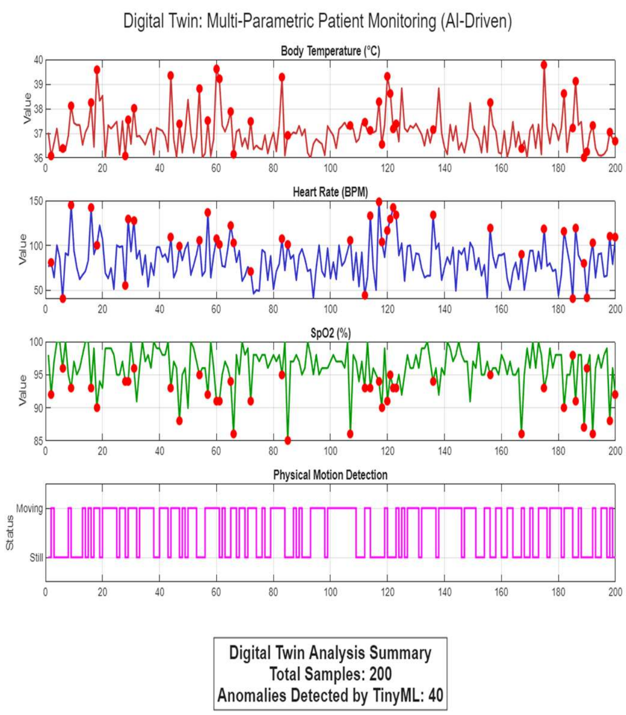


Fig 5.1.2 Classification

4.Edge decision function

The edge decision function controls the transfer of data depending on the results obtained from the TinyML classification. If the analysis of the sensor data detects that there is no anomaly in the patient's condition, the system suppresses the transfer of data and puts the sensor node into sleep mode to save energy. On the other hand, if the analysis detects any anomaly in the patient's condition, the system initiates the transfer of data to the gateway.

Temperature	Heart Rate	SpO2	Motion	Edge AI Decision	Action
36.8	75	98	0	Normal	No Alert
37.5	82	97	1	Normal	No Alert
39	130	88	0	⚠ Abnormal	Alert Sent
35.8	48	94	0	⚠ Abnormal	Alert Sent
38.2	115	89	1	⚠ Abnormal	Alert Sent



5.2 Digital twin result

In the MATLAB platform, the Digital Twin algorithm is simulated to show the real-time performance of the patient monitoring system by using inputs from data sets. The digital twin acts as an alternative to the actual patient monitoring system, showing the parameters like temperature, heartbeat rate, and SpO_2 level on graphs. The use of digital twin will help in the clear understanding of the patient's condition.

The Digital Twin is a reflection of the way the decision-making system works in the TinyML environment. In the case where there are no anomalies detected in the sensor readings, the system does not send any data or sends minimal data, which can be seen in the simulations. However, when there are anomalies detected, the system immediately sends data, proving its fast response to emergencies.

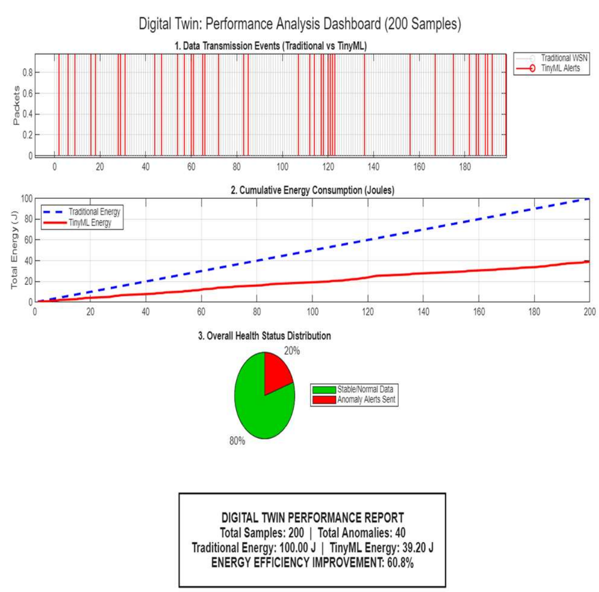
Moreover, it is easier to perform the analysis of the performance of the system using Digital Twin, which includes analyzing the transmission modes and energy efficiency of the system without the need for any physical setup. Various scenarios can be performed in MATLAB to check how the system performs under certain conditions. In conclusion, it can be said that the use of Digital Twin model improves the performance of the healthcare monitoring system.

5.3 Compare Normal WSN and TinyML WSN

In the case of traditional Wireless Sensor Network (WSN), the nodes send all gathered information constantly to the server regardless of the nature of the data, be it normal or abnormal. Therefore, such a network consumes more energy, causing network traffic and shortening the lifetime of the sensor node. Moreover, as the network lacks intelligence at the edge, all computations are centralized, adding more latency.

However, the WSN employing TinyML employs edge intelligence by incorporating a lightweight ML model within the sensor node itself. In place of sending the data to the network, the device first evaluates the sensor readings and categorizes them into normal and abnormal. Data is sent only if there is any abnormality present; otherwise, normal data is filtered out.

Conclusively, the performance of TinyML-based WSN is better than that of conventional WSN because it consumes less energy, has less data transmission, makes quick decisions, and has improved efficiency and accuracy.



6. Conclusion

This paper provides a new patient monitoring solution that employs the concept of WSNs, TinyML, and Digital Twins to overcome the drawbacks of existing healthcare monitoring methodologies. The use of TinyML in the design makes it possible to carry out data processing locally at the sensor nodes. Thus, there is no wastage of energy through unnecessary data transmission, which greatly enhances energy efficiency. Selective communication is used to ensure that only important data concerning the patient's health status is transmitted over the network.

The model of the Digital Twin created in MATLAB enhances the system through a simulation platform that can be used to study the actual workings of the system in a virtual environment without deploying the physical system. The model makes it easier to validate the performance of the system.

Comparative study of the conventional WSN model and the TinyML-based WSN reveals that the new approach provides high efficiency, high speed, and higher reliability. In general, the adoption of edge intelligence and virtualization technology makes the design scalable, economical, and appropriate for contemporary smart health care solutions.

References

[1] A. Adeli, N. Bagheri, and J. J. Rodrigues, "A Post-Quantum Authentication Scheme for IoT Healthcare Systems," *IEEE Internet of Things Journal*, 2023.

[2] M. Al-Sadoon, A. Jedidi, and H. Al-Raweshidy, "Dual-Tier Cluster-Based Routing in Mobile Wireless Sensor Networks for IoT Applications," *IEEE Access*, vol. 11, pp. 4079–4094, 2023.

[3] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[4] F. Chollet, *Deep Learning with Python*. Shelter Island, NY, USA: Manning Publications, 2017.

[5] A. K. Jameil and H. Al-Raweshidy, "A Digital Twin Framework for Real-Time Healthcare Monitoring: Leveraging AI and Secure Systems for Enhanced Patient Outcomes," *Discover Internet of Things*, vol. 5, no. 1, pp. 1–15, 2025.

[6] Z. Jia *et al.*, "TinyML Design for Life-Threatening Arrhythmia Detection on Low-Power Devices," in *Proc. IEEE/ACM ICCAD*, 2023.

[7] F. T. Liu, K. M. Ting, and Z. Zhou, "Isolation Forest," in *Proc. IEEE International Conference on Data Mining (ICDM)*, 2008, pp. 413–422.

[8] A. Rahmani, N. K. Thanigaivelan, T. Nguyen, D. C. Nguyen, and P. Liljeberg, "Smart e-Health Gateway: Bringing Intelligence to Internet-of-Things-Based Ubiquitous Healthcare Systems," *IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 826–834, 2015.

[9] V. Rishiwal, U. Agarwal, M. Yadav, S. Tanwar, D. Garg, and M. Guizani, “Machine Learning and Quantum Computing for IoT Networks: Concepts and Challenges,” *IEEE Internet of Things Journal*, 2025.

[10] S. Raza, P. Misra, Z. He, and T. Voigt, “Building the Internet of Things with Low Power and Lossy Networks,” *IEEE Internet Computing*, vol. 21, no. 2, pp. 10–17, Mar.–Apr. 2017.

[11] M. Satyanarayanan, “The Emergence of Edge Computing,” *IEEE Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017.

[12] F. Tao, Q. Qi, L. Wang, and A. Y. C. Nee, “Digital Twins and Cyber–Physical Systems Toward Smart Manufacturing and Industry 4.0: Correlation and Comparison,” *Engineering*, vol. 5, no. 4, pp. 653–661, 2019.

[13] T.-H. Vu, N. H. Tu, T. Huynh-The, K. Lee, S. Kim, M. Voznak, and Q.-V. Pham, “Integration of TinyML and LargeML for 6G and Beyond: A Survey,” 2025.

[14] T. Watt, C. Chrysoulas, and P. J. Barclay, “Moving Healthcare AI-Support Systems onto Constrained Devices Using TinyML,” 2024.

[15] P. Warden and D. Situnayake, *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. Sebastopol, CA, USA: O’Reilly Media, 2019.

[16] G. Yang, L. Xie, M. Mäntysalo, X. Zhou, Z. Pang, L. Da Xu, S. Kao-Walter, Q. Chen, and L. R. Zheng, “A Health-IoT Platform Based on the Integration of Intelligent Packaging, Unobtrusive Bio-Sensor, and Intelligent Medicine Box,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2180–2191, Nov. 2014.

[17] A. Zila, “Edge AI and TinyML for Enhancing MAC Protocols in Wireless Sensor Networks,” *International Journal of Communication Systems*, 2026 (early access 2025).