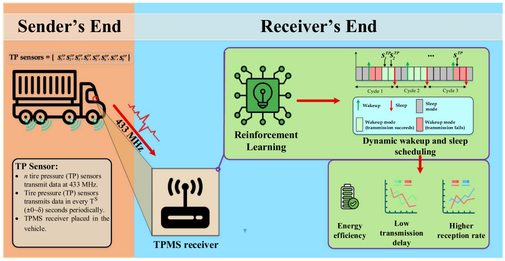
Dynamic Sleep Scheduling for Wireless TP Sensor Transmissions Based on Reinforcement Learning

Shashank Mishra and Jia-Ming Liang*

Department of Electrical Engineering, National University of Tainan, Tainan 700301, Taiwan
**Member, IEEE*

Manuscript received 31 August 2023; revised 25 September 2023; accepted 12 October 2023. Date of publication 23 October 2023; date of current version 7 November 2023.

Abstract—*Tire Pressure (TP)* sensors are an essential component of the *Tire Pressure Monitoring System (TPMS)* for smart vehicles. Each TP sensor is mounted on a wheel for real-time detecting and monitoring abnormal tire conditions to ensure driving safety. In the TPMS system, multiple TP sensors transmit sensing data randomly at a regular period to the TPMS receiver with a special communication module of 433 MHz. Traditionally, the TPMS receiver adopts a power saving mechanism to perform wake-up and sleep operation within a fixed cycle length for sensing data collection to save energy. However, the transmission delay may become worse once the number of TP sensors increases. Therefore, this letter tries to consider such problem and develops a dynamic scheduling approach for the TPMS by balancing the energy efficiency while ensuring data transmission delay. Specifically, the proposed approach exploits a reinforcement learning approach to dynamically control the wake-up and sleep periods of the TPMS receiver based on the designed reward function. Simulation results show that the proposed approach can significantly decrease data transmission delay by 21.94%–64.13% in average and improve energy efficiency by 5.29%–39.57% in average, as compared with the existing schemes.


Sender's End
Receiver's End
TP sensors = { $S_1^{TP}, S_2^{TP}, S_3^{TP}, S_4^{TP}, S_5^{TP}, S_6^{TP}, S_7^{TP}, S_8^{TP}$ }
433 MHz
TP Sensor:
n tire pressure (TP) sensors transmit data at 433 MHz.
Tire pressure (TP) sensors transmits data in every T^S (±0~δ) seconds periodically.
TPMS receiver placed in the vehicle.
TPMS receiver
Reinforcement Learning
S_1^{TP} S_2^{TP} ... S_8^{TP}
Cycle 1
Cycle 2
Cycle 3
Wakeup
Sleep
Sleep mode
Wakeup mode (transmission succeeds)
Wakeup mode (transmission fails)
Dynamic wakeup and sleep scheduling
Energy efficiency
Low transmission delay
Higher reception rate

Index Terms—Sensor Systems, dynamic scheduling, energy efficiency, reinforcement learning, tire pressure (TP) sensor transmission, transmission delay.

I. INTRODUCTION

Tire Pressure Monitoring System (TPMS) is an advanced technology designed to monitor abnormal conditions and ensure driving safety [1]. The system relies on *tire pressure (TP)* sensors, which are mounted on a vehicle's tires, enabling to transmit real-time data wirelessly at regular intervals to the TPMS receiver through a dedicated module operating at 433 MHz, as shown in Fig. 1.

In a TPMS, transmission delay and energy efficiency are critical factors. Timely transmission from TP sensors is crucial for real-time monitoring and prompt detection of abnormal tire conditions, as delays can compromise vehicle safety and performance [2]. Energy efficiency directly affects TPMS battery life, optimizing sleep operation and minimizing the need for frequent battery replacements or recharging [3], [4]. Traditionally, the TPMS receiver adopts a power saving mechanism, which performs wake-up and sleep operations within a fixed cycle length to collect sensing data so as to improve energy efficiency [5]. However, if the number of TP sensors increases, such as the truck with six TP sensors or container car with 12 TP sensors, as shown in Fig. 1, the transmission delay would become worse severely. Therefore, this letter tries to address these issues and asks how to optimize the power saving mechanism of the TPMS receiver to balance the tradeoff between transmission delay and energy efficiency.

To tackle this problem, we propose a dynamic sleep scheduling for the TPMS based on reinforcement learning technology. The main idea of reinforcement learning is to train an agent based on the current data arrival rate to make optimal decisions in an environment, specifically estimating the optimal wake-up and sleep actions, which can potentially reduce the transmission delay and prioritize energy efficiency. Through extensive simulations, it shows that the proposed scheme can improve the energy efficiency and transmission delay, as compared with traditional schemes.

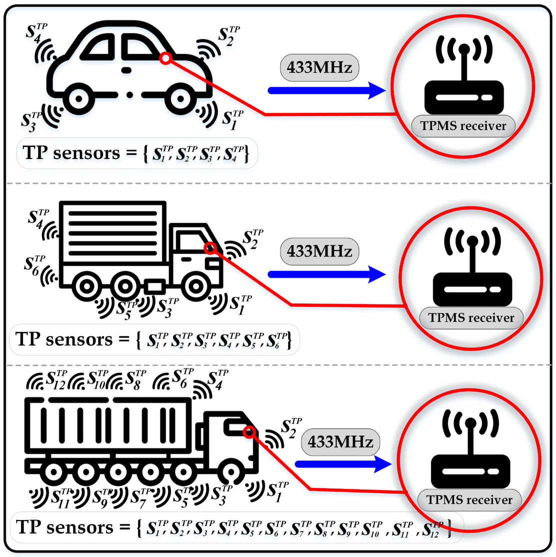

S_4^{TP}
S_2^{TP}
S_3^{TP}
S_1^{TP}
433MHz
TPMS receiver
TP sensors = { $S_1^{TP}, S_2^{TP}, S_3^{TP}, S_4^{TP}$ }
S_4^{TP}
S_6^{TP}
S_2^{TP}
S_5^{TP} S_3^{TP} S_1^{TP}
433MHz
TPMS receiver
TP sensors = { $S_1^{TP}, S_2^{TP}, S_3^{TP}, S_4^{TP}, S_5^{TP}, S_6^{TP}$ }
S_{12}^{TP} S_{10}^{TP} S_8^{TP} S_6^{TP} S_4^{TP}
S_2^{TP}
S_{11}^{TP} S_9^{TP} S_7^{TP} S_5^{TP} S_3^{TP} S_1^{TP}
433MHz
TPMS receiver
TP sensors = { $S_1^{TP}, S_2^{TP}, S_3^{TP}, S_4^{TP}, S_5^{TP}, S_6^{TP}, S_7^{TP}, S_8^{TP}, S_9^{TP}, S_{10}^{TP}, S_{11}^{TP}, S_{12}^{TP}$ }

Fig. 1. TPMS with (a) four TP sensors, (b) six TP sensors, and (c) 12 TP sensors.

Corresponding author: Jia-Ming Liang (e-mail: jmliang@mail.nutn.edu.tw).
Associate Editor: D. Popescu.
Digital Object Identifier 10.1109/LSENS.2023.3327002

II. RELATED WORK

In recent years, numerous approaches have been made to address the importance of energy efficiency and transmission delay for wireless sensors [6], [7]. To consider these issues, several approaches have been proposed [5], [8], [9]. In [5], a power saving mechanism was adopted to perform wake-up and sleep for data collection considering the fixed-sleep cycle length. In [8], a tree construction and link scheduling approach was proposed, whereas [9] adopted a breadth-first search approach to reduce the energy consumption. However, these studies primarily focused on energy-saving strategies with a limited number of sensor nodes, neglecting the challenges that emerge when handling a larger number of sensor nodes.

The authors of [10], [11], and [12] presented algorithms that focused on reducing transmission delay in wireless sensors through elliptic curve, dynamic adjustment, and wake-up scheduling techniques, respectively. However, these studies neglected the significance of energy efficiency in addition to minimizing data transmission delay. The authors in [13] and [14], proposed approaches to enhance energy efficiency in wireless sensors through reinforcement learning, optimal sleep scheduling, and stochastic cooperative decision, respectively. However, these studies primarily focused on energy efficiency and did not comprehensively consider data transmission delay. The authors in [15] and [16] introduced energy-efficient approaches for collecting periodic and real-time data in wireless sensors and nanosensor networks, respectively. However, these studies emphasize energy efficiency and neglect the comprehensive consideration of transmission reception ratio and data transmission delay. This motivates us to study the problem.

III. PROBLEM DEFINITION

In this letter, we consider a TPMS system with one receiver and n TP sensors denoted by $S^{\text{TP}} = \{s_1^{\text{TP}}, s_2^{\text{TP}}, \ldots, s_n^{\text{TP}}\}$. Each TP sensor periodically transmits its sensing data through a 433 MHz module at an interval of $T^S(\pm 0 \sim \delta)$ s to the TPMS receiver and forms a star topology [17]. In addition, the TPMS receiver can be scheduled to wake up and receive the TP sensing data or go to sleep to save energy. Our problem asks how to well determine the wake-up and sleep periods of the system to enhance energy efficiency, minimize transmission delay, while increasing the data reception rate of TP sensor transmissions.

IV. PROPOSED SCHEME

In this section, we will introduce a novel scheduling approach, which leverages a Q-learning based reinforcement learning approach to dynamically perform wake-up and sleep operation. The main idea is to monitor the historic data reception rate and estimate the current data arrival possibility to perform wake-up and sleep operations, which can potentially reduce the transmission delay and ensure energy efficiency. Specifically, the Q-learning composes of the *state* and *action* spaces, and trains an *agent* to learn and maximize the *reward* value. The *agent* dynamically performs the actions of wake-up and sleep based on the learned Q-values, where the training process iteratively updates Q-values according to the observed rewards. The concept of Q-learning is illustrated in Fig. 2.

In the following, we precisely describe the details of each component of Q-learning.

A. State Space

We define the state space as $S = \{s_1, s_2, .., s_t, ..\}$, which represents the system conditions at time slot t, including historic data reception rate, energy efficiency of system, and transmission delay history. The transmission delay history refers to the record of past transmission delays of each sensor experienced in the system. It captures the time delays observed during previous transmissions and provides a historical context for the current system conditions.

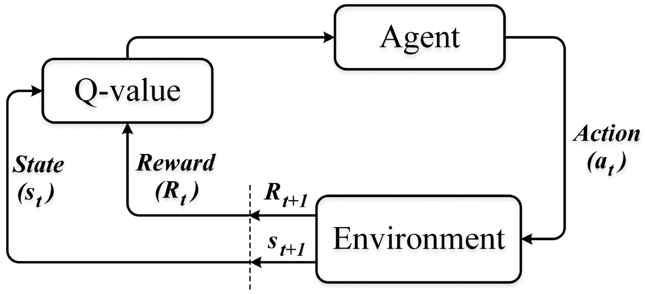


Fig. 2. Concept of Q-learning approach.

B. Action Space

We define the *action* space as $A = \{a_1, a_2, \ldots, a_t, ..\}$, where $a = 1$ is for wake-up and $a = 0$ is for sleep, which represents the set of possible actions that the *agent* can take. In our proposed approach, the actions correspond to the scheduling decisions for the TPMS receiver.

C. Agent

The agent learns to make the best scheduling decision by observing the current state (wake-up or sleep) of the system and selecting actions that maximize the expected cumulative reward over time.

D. Q-Learning Approach

We utilize the Q-learning algorithm [11] to train the agent. The Q-learning algorithm maintains a Q-value function, $Q(s, a)$, which estimates the expected cumulative reward when taking action a at state s. The Q-value function is updated iteratively based on the observed rewards and the learning rate (α).

E. Reward Function

By designing the reward function and using the Q-learning, our proposed approach can dynamically schedule the wake-up and sleep for the TPMS receiver. The reward function is defined as follows:

$$R(s_t, a_t, s_{t+1}) = \omega_s \times R_R(n, t) + \omega_e \times E_E(n, t) - \omega_d \times T_D(n, t) \quad (1)$$

where $R(s_t, a_t, s_{t+1})$ represents the reward function and ω_s, ω_e, and ω_d are the weighting factors to determine the relative importance of successful data reception rate $R_R(n, t)$, energy efficiency $E_E(n, t)$, and transmission delay $T_D(n, t)$, respectively. Specifically, this reward function takes into account factors of successful data reception rate $R_R(n, t) = \frac{\sum_{i=1}^n N_i^R(t-W^S:t)}{N^T(t-W^S:t)}$, where $\sum_{i=1}^n N_i^R(t - W^S : t)$ represents the number of TP sensors received, and $N^T(t - W^S : t)$ represents the total arriving TP sensors during a window size W^S. Energy efficiency is defined by $E_E(n, t) = \frac{\sum_{i=1}^t W_i^R(t-W^S:t)}{W^T(t-W^S:t)}$, where $\sum_{i=1}^t W_i^R(t - W^S : t)$ is the total number of wake-up slots with receiving transmission and $W^T(t - W^S : t) = \sum_{i=(t-W^S)}^t a_t$ are the total wake-up slots during the same interval. Transmission delay is defined by $T_D(n, t) = [\frac{\sum_{i=1}^n (T_i^R - T_i^{R'}) - T^S}{n}]^+$, where $T_i^{R'}$ is the TP sensor successful received time previously, and T_i^R is TP sensor received time after $T_i^{R'}$ in the TPMS system. If this difference exceeds a sensor arriving interval T^S, it will be counted as a transmission delay.

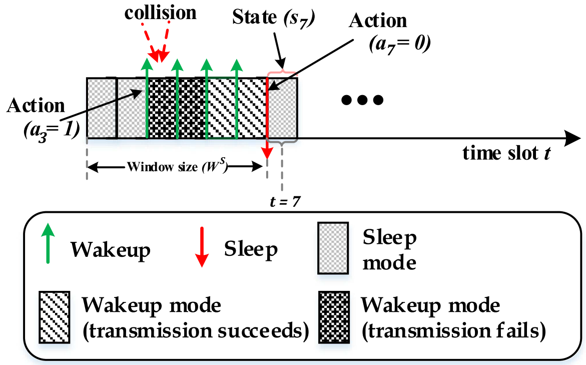


Fig. 3. Dynamic sleep scheduling of the TPMS with Q-learning.

F. Q-Learning Model Training

A. Initialization: Initialize the Q-table with arbitrary values for each state–action pair.

B. Training Loop: We iterate through the following steps until convergence.

1) Monitor the *state* of TPMS receiver, denoted by s.
2) Observe the current *state* at time slot t of TPMS receiver, denoted by s_t.
3) Choose an *action* of time slot t, a_t (wake-up or sleep), for the TPMS receiver based on the epsilon-greedy policy using the Q-values.
4) Execute the selected *action* a_t, and observe the resulting *state* s_{t+1}, including energy efficiency $E_E(n,t)$, and transmission delay $T_D(n,t)$, and successful data reception rate $R_R(n,t)$, based on the current state (s_t), selected action (a_t), and resulting state (s_{t+1}).
5) Update the Q-value of the chosen state–action pair using the Q-learning update equation

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[R(s_t, a_t, s_{t+1}) + \gamma \max(Q(s_{t+1}, a_{t+1})) - Q(s_t, a_t)] \quad (2)$$

where $Q(s_t, a_t)$ represents the Q-value of state–action pair (s_t, a_t), and $R(s_t, a_t, s_{t+1})$ denotes the reward received for taking action a_t at state s_t to state s_{t+1}. Note that α is the learning rate (where $0 < \alpha < 1$) and γ is discount factor (where $0 \leq \gamma < 1$).

G. Q-learning based Dynamic Scheduling

1) Deploy the trained Q-learning model in the system to dynamically schedule the current slot to wake-up or sleep in the TPMS receiver.
2) Maximize the *reward* function, denoted by $R(s_t, a_t)$, through the predefined performance metrics in terms of sensor reception rate, energy efficiency, and transmission delay.

At the end of the training process, we obtain the final Q-table, which represents the learned optimal action values for each state–action pair.

In the following, we show an example of our approach with Q-learning in Fig. 3, to calculate the value of reward function based on the three important factors, where the current time slot is $t = 7$ and there are $n = 4$ TP sensors. In this figure, we can see that it has performed sleep for the first two slots and wake-up for the remaining four slots. Due to transmission delay or collision, two wake-up slots failed to receive the transmission, whereas next two wake-up slots successfully received the transmissions. According to (1), the successful data reception rate $R_R(4,7) = \frac{2}{4} = 0.5$, means only two TP sensors received over a total of four TP sensor transmissions; energy efficiency $E_E(4,7) = \frac{2}{4} = 0.5$, means only two wake-up slots with receiving transmission over a total of four wake-up slots. Now, considering the weighting factors are $\omega_s = \omega_e = 0.4$ and $\omega_d = 0.2$, the reward value is derived by $R(s_7, a_7, s_8) = 0.4 \times \frac{2}{4} + 0.4 \times \frac{2}{4} - 0.2 \times 0 = 0.4$ accordingly. Note that $T_D(4,7) = 0$ as we assume the transmission delay does not exceed the sensing arriving interval.

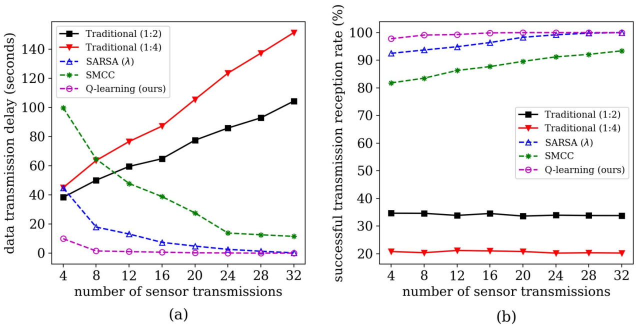


Fig. 4. Comparisons on (a) data transmission delay and (b) successful transmissions reception rate of all schemes.

V. PERFORMANCE EVALUATION

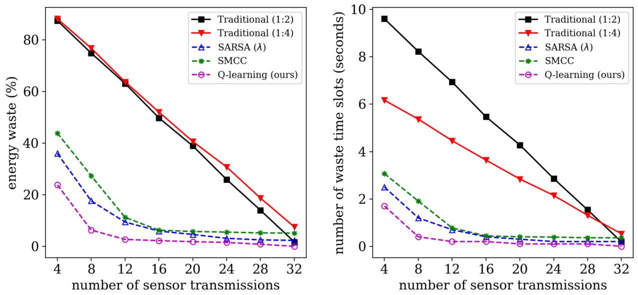
In this section, we conduct a simulator by Python to evaluate the efficiency of our proposed scheme. The simulation includes 4–32 TP sensors with a periodic and random transmission at an interval of $T^S(\pm 0\text{–}\delta)$ seconds, where $T^S = 32$ and $\delta = 2$ [17]. We compare our scheme against a *Traditional (1:2)*, *Traditional (1:4)* [5], *SARSA*(λ) [16], and *SMCC* [14] schemes. The traditional scheme uses a fixed cycle length to perform wake-up and sleep operation to ensure the transmission delay while saving energy, where the sleep cycle length is 3 s (1 s for wake-up and 2 s for sleep) and 5 s (1 s for wake-up and 4 s for sleep) denoted as *Traditional (1:2)* and *Traditional (1:4),* respectively. *SARSA*(λ) is a temporal-difference learning-based algorithm, which updates action-value estimates by considering the immediate *reward*. The *SMCC* scheme enables sensors to determine their optimal cooperation scope using stochastic decision-making considering metrics for energy and delay. In the simulation, the experiment is examined by >10 000 time slots. Note that the weighting factors of our scheme are $\omega_s = \omega_e = 0.4$ and $\omega_d = 0.2$. In addition, we use learning rate $\alpha = 0.9$, discount factor $\gamma = 0.1$, and window size $W^S = 32$.

A. Data Transmission Delay

First, we observe the data transmission delay of all schemes. In Fig. 4(a), we can see that our *Q-learning* approach outperforms the *Traditional (1:2)*, *Traditional (1:4)*, *SARSA*(λ), and *SMCC* schemes for any number of sensor transmissions. The *Traditional (1:2)* and *Traditional (1:4)* schemes use fixed wake-up and sleep slots, whereas *SARSA(λ)* and *SMCC* update values based on the actual actions taken during exploration, which causes slower convergence and higher transmission delays as the number of sensor transmissions increases. Our approach dynamically adjusts the wake-up and sleep periods based on learned Q-values to optimize the cycle length and ensure timely data transmission. This significantly decreases data transmission delays by an average of 21.94%–64.13% compared with other schemes.

B. Transmission Reception Ratio

Next, we observe successful transmission reception ratio of all schemes. In Fig. 4(b), our *Q-learning* approach consistently outperforms *Traditional (1:2)*, *Traditional (1:4)*, *SARSA*(λ), and *SMCC* schemes across any numbers of sensor transmissions. The *Traditional*



(a)

(b)

Fig. 5. Comparisons on (a) energy waste and (b) number of waste time slots of all schemes.

(1:2) and *Traditional (1:4)* schemes rely on fixed cycle lengths for wake-up and sleep operations over random sensor transmission, resulting in lower reception rate, whereas *SARSA*(λ) and *SMCC* schemes are more sensitive to policy stability, environmental changes, and stochastic cooperative decisions, respectively. As a result, these schemes lead to lower transmission reception ratio. The proposed *Q-learning* approach effectively explores different wake-up and sleep schedules for better transmission reception ratios. Since *Q-learning* learns from the best actions, regardless of the agent's behavior, enabling the identification of optimal sleep patterns, adaptation to system changes, and optimization of transmission reception timings. Our approach ultimately improves transmission reception ratios.

C. Energy Waste

We observe the energy waste of all schemes based on window size $W^S = 32$, as shown in Fig. 5(a). The energy waste is defined as Energy Waste(%) $= (1 - E_E(n,t)) \times 100$. We can see that our *Q-learning* approach surpasses *Traditional (1:2)*, *Traditional (1:4)*, *SARSA*(λ), and *SMCC* schemes in terms of energy waste through dynamic sleep scheduling, optimal decision-making, balanced exploration-exploitation, and delayed reward learning. The *Traditional (1:2)* and *Traditional (1:4)* schemes use fixed wake-up and sleep slots, which are inefficient for varying sensor transmission. Meanwhile, *SARSA(λ)* exhibits slow convergence and makes suboptimal decisions, and *SMCC* leads to suboptimal sleep scheduling decisions in dynamic scenarios, resulting in increased energy waste. Our scheme adaptively adjusts sleep intervals and considers both short-term, the immediate benefits from specific actions and long-term rewards, the cumulative benefits over the time, resulting in reduced energy consumption by 5.29%–39.57% on average.

D. Number of Waste Time Slots

Finally, considering a window size of $W^S = 32$, we examine number of waste time slots across all methods, as depicted in Fig. 5(b). The number of waste time slots is defined as the total wake-up slots without receiving transmission data over a window size. Notably, our *Q-learning* approach outperforms all other schemes in terms of the number of waste time slots. The superiority of *Q-learning* over *Traditional (1:2)*, *Traditional (1:4)*, *SARSA*(λ), and *SMCC* schemes is achieved through its adaptive capabilities and dynamic sleep scheduling. The *Traditional (1:2)* and *Traditional (1:4)* schemes use a fixed number of wake-up slots, which is not optimal for randomly transmitted sensors. In contrast, *SARSA*(λ) adopts an on-policy strategy, and *SMCC* selects task schedules aimed at achieving comprehensive decisions, ultimately leading to an increase in the number of wasted time slots.

VI. CONCLUSION

In this letter, we have addressed the sleep scheduling problem of the TPMS under the consideration of energy efficiency, transmission delay, and the successful data reception rate. We have introduced a dynamic scheduling approach based on Q-learning. The main idea is to optimize wake-up and sleep operations based on the designed reward function to potentially reduce energy consumption while maintaining transmission reception ratio. Simulation results have verified that our approach has better performance in terms of energy efficiency and transmission reception ratio.

ACKNOWLEDGMENT

This work was supported in part by NSTC under Grant 109-2221-E-024-012-MY3 and Grant 109-2221-E-024-013-MY3.

REFERENCES

[1] B. Afshar, M. Fathy, M. Asgari, M. Shahverdy, and P. Shahverdi, "A machine learning-based approach to detect polluting vehicles in smart cities," in *Proc. 6th Int. Conf. Smart Cities, Internet Things Appl.*, 2022, pp. 1–5.

[2] S. Manikandan, N. Poongavanam, V. Vivekanandhan, and T. A. Mohanaprakash, "Performance comparison of various wireless sensor network dataset using deep learning classifications," in *Proc. IEEE Int. Conf. Mobile Netw. Wireless Commun.*, 2022, pp. 1–4.

[3] S. M. A. Rahaman and M. Azharuddin, "An efficient charging scheduling scheme to enhance the wireless rechargeable sensor networks' lifespan," in *Proc. Int. Conf. Intell. Syst. Adv. Comput. Commun.*, 2023, pp. 1–6.

[4] M. I. Tawakal, M. Abdurohman, and A. G. Putrada, "Wireless monitoring system for motorcycle tire air pressure with pressure sensor and voice warning on helmet using fuzzy logic," in *Proc. Int. Conf. Softw. Eng. Comput. Syst. 4th Int. Conf. Comput. Sci. Inf. Manage.*, 2021, pp. 47–52.

[5] S. Mishra and J.-M. Liang, "Design and analysis for wireless tire pressure sensing system," in *Proc. Int. Comput. Symp.*, 2022, pp. 599–610.

[6] X. Luo, C. Chen, C. Zeng, C. Li, J. Xu, and S. Gong, "Deep reinforcement learning for joint trajectory planning, transmission scheduling, and access control in UAV-assisted wireless sensor networks," *Sensors*, vol. 23, no. 10, 2023, Art. no. 4691.

[7] X. Fu and J. G. Kim, "Deep-q-network-based packet scheduling in an IoT environment," *Sensors*, vol. 23, no. 3, 2023, Art. no. 1339.

[8] N. Ghosh and I. Banerjee, "Energy-efficient compressive sensing based data gathering and scheduling in wireless sensor networks," *Wireless Pers. Commun.*, vol. 128, pp. 2589–2618, 2023.

[9] B. R. Srinivasa, "Energy and delay balance ensemble scheduling algorithm for wireless sensor networks," in *Proc. Res. Adv. Netw. Technol.*, 2023, pp. 1–14.

[10] B. Bettoumi and R. Bouallegue, "Efficient reduction of the transmission delay of the authentication based elliptic curve cryptography in 6LoWPAN wireless sensor networks in the Internet of Things," in *Proc. Int. Wireless Commun. Mobile Comput.*, 2021, pp. 1471–1476.

[11] J. Li, W. Shi, N. Zhang, and X. Shen, "Delay-aware VNF scheduling: A reinforcement learning approach with variable action set," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 1, pp. 304–318, Mar. 2021.

[12] J. Mhatre and A. LeeG, "Dynamic reinforcement learning based scheduling for energy-efficient edge-enabled LoRaWAN," in *Proc. IEEE Int. Performance, Computing, Commun. Conf.*, 2022, pp. 412–413.

[13] X. Cao, J. Wang, Y. Cheng, and J. Jin, "Optimal sleep scheduling for energy-efficient AoI optimization in industrial Internet of Things," *IEEE Internet Things J.*, vol. 10, no. 11, pp. 9662–9674, Jun. 2023.

[14] Y. Yang and T. Song, "Energy-efficient cooperative caching for information-centric wireless sensor networking," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 846–857, 2022.

[15] S. Kumar and H. Kim, "Energy efficient scheduling in wireless sensor networks for periodic data gathering," *IEEE Access*, vol. 7, pp. 11 410–11 426, 2019.

[16] J. Xu, H. Huang, Y. Zhao, and L. Lin, "A MAC protocol based on energy scheduling for in-vivo wireless nanosensor networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2021, pp. 1–6.

[17] AVE, "Ave external sensor specification," Accessed: Sep. 23, 2023. [Online]. Available: http://www.avetechnology.com/index.php?unit=products&lang=en&act=view&id=89