*International Journal of Basic and Applied Sciences*, 14 (7) (2025) 414-423

International Journal of Basic and Applied Sciences

*Website: www.sciencepubco.com/index.php/IJBAS*
*https://doi.org/10.14419/q2va4b09*
***Research paper***

# A Power-Saving Classification Module for The Internet of Things Enclosed Wireless Sensors Embedded in Smart Controllers

**Dr. R. Gayathri [1] *, Dr. P. Gnanasundari [2], Dr. S. Padmanaban [3], Dr. B. Suganthi [4], Poornima R. [5], Prof. Nitin Chakole [6]**

*[1] Assistant Professor, Department of Electronics and Communication Engineering, Karpagam Academy of Higher Education, Coimbatore,*
*[2] Associate Professor, RV University, Bangalore*
*[3] Professor, Department of Mathematics - Science and Humanities, Karpagam Institute of Technology, Coimbatore*
*[4] Associate Professor, Department of ECE, RVS College of Engineering and Technology, Coimbatore*
*[5] Assistant Professor, Department of ECE, SNS College of Technology, Coimbatore*
*[6] Assistant Professor, Department of Electronics and Communication Engineering, Ramdeobaba University, Nagpur,*
**Corresponding author E-mail: gayathri12121504@gmail.com*

*Received: September 11, 2025, Accepted: November 10, 2025, Published: November 15, 2025*

## Abstract

Internet of Things (IoT), combined with wireless sensor network (WSN) devices, interacts with smart controllers within industries or buildings, etc., to accomplish field jobs. The jobs/ tasks are completed based on the controller's decision, power-saving, and sustainable operation intervals. To improve the power savings of such devices, a Control-dependent Process Classification Module (CdPCM) is proposed in this article. This proposed module balances the controller operation time and the number of input controls based on a low-power sustainable feature. The feature identifies the maximum operation time of the controller before reaching the recharging state. In this process, the IoT layers are responsible for commissioning and decommissioning jobs for the low-power operations. Thus, the input controls are relocated from the overloaded controllers to sustain their active state until an alternate controller is reassigned. Thus, the power-saving schedules are allocated for the sensor-based controllers to accomplish the tasks with minimal inactive time. The factor classification based on the above feature is recurrent until the controller operation interval is active, using deep recurrent learning networks. The proposed module improves power saving by 11.22%, reduces the power consumption by 11.03% and idle power loss by 10.67% for the varying job intervals.

***Keywords***: *DRL; IoT; Power Saving; Process Allocation; Smart Controllers.*

## 1. Introduction

Power conservation mechanisms are vital for wireless sensor-driven embedded controllers, particularly in energy-limited scenarios. Such systems tend to be based on battery-powered nodes, and therefore, efficient use of energy is key to long-term deployment [1], [2]. Hardware optimizations at the node level, such as ultra-low-power microcontrollers and power-efficient transceivers, also play a critical role [3]. Data transmission, which is the most energy-consuming operation, tends to be minimized by local processing and compression [4], [5].

Internet of Things (IoT) infrastructure increases power monitoring and management in embedded controllers through wireless sensors [6]. Real-time data capture enables dynamic analysis of energy utilization patterns through different components [7]. IoT platforms can change the behavior of the controller depending on the intensity of the workload and environmental factors to save energy [8]. Historical energy profiles are optimized to maximise future energy utilization through cloud integration [9].

Machine learning (ML) techniques facilitate smart power-saving decisions in wireless sensor-controlled embedded systems [10]. Algorithms are able to learn the best patterns of power consumption from historical information, anticipating when sensors should be turned on or off [11], [12]. Reinforcement learning methods are well-suited for constant learning from evolving conditions and workloads. Classifiers assist in the detection of energy-intensive operations, with real-time adjustments made to reduce their occurrence [13], [14]. The contributions of the article are:

1) The proposal and description of a control-dependent process classification module to address the power-saving constraints, ensuring small IoT-based controller operations, is given.
2) This proposed module uses deep recurrent learning to perform classification on power-saving and consuming schedules. The schedules are used to balance the task completion through power loss reduction.
3) The proposed module is analyzed using experimental validation and comparative analysis to verify its efficiency under different variants.

## 2. Related Works

To decrease the use of power during idle time, Liu et al. [15] proposed the wake-up signal-based discontinuous reception control method. The incoming signal triggers the system to avoid unnecessarily activating the radio. The timing model calculates the delay to keep the data in sleeping cycles. The period of DRX varies dynamically in correspondence with traffic and user activity.

Nsaif et al. [16] proposed a framework of link prediction in software-defined data centers. The inactive transmission paths are detected by applying an artificial neural network model for the power-saving effect. The traffic flow data is monitored constantly and routed by means of efficient routes. Power consumption gets lower while keeping performance quite stable.

To have effective indoor light control, Singh et al. [17] devised a visible light communication-based LED power optimization model. The method detects physical obstructions and redirects light only to clear zones within rooms. Power allocation is optimized so that only effective signal zones receive illumination. It avoids wasting energy on lighting that is not in use or is blocked from use.

Wang et al. [18] introduced a proportional-integral-derivative controller for better thermal regulation of greenhouses. The actuator is capable of controlling a heating system that predicts the requirements for heating through live sensor information and environmental changes. The method maintains a stable warm condition due to the automatic adjustment of the heating. The controller prevents unwanted fuel consumption with exact precision.

Rao et al. [19] presented a smart scheduling system to manage electricity loads in homes. The model shifts high-consumption appliance operations away from peak electricity hours. Requirements and comfort levels are considered from the user side while reducing the total draw. The system reacts to live grid changes, effectively relating load and demand.

Emami et al. [20] proposed a hybrid energy model wherein the fuel cells are combined with demand forecasting to balance energy usage in smart buildings. Futures load requirements are estimated by energy demand based on the electric price, weather trend, and the occupancy data. Surplus energy is stored and reused during high-demand hours to lessen peak stress. The system saves money on electricity without compromising thermal comfort.

To reduce power consumption for cooling at home, Sirisumrannukul et al. [21] developed an air-conditioning control model applying artificial neural networks and particle swarm optimization. Fan speed and airflow direction will be modified according to the heat load and occupant comfort preferences. The model uses room sensor inputs to change the temperature to the required level in real time. The automated control is capable of changing the conditions without the user's intervention.

Zhao et al. [22] proposed a control architecture using fuzzy logic and quantum-inspired decision-making. The use of appliances is scheduled differently across historical behavior as well as current energy demands. Intelligent control units take care of load adjustments while addressing user comfort and utility pricing. Adaptive smart scheduling mechanisms are used to deal with system-level constraints.

Pandey et al. [23] introduced a flexible lightning control option aided by IoT for smart living environments. This mode assimilated control, light, and remote sensors to enhance the dynamic lighting control from remote locations. Such ambient lighting control is introduced to enhance the power-saving nature of the environment.

To enhance the lifetime of infrared sensors, Zhong et al. [24] proposed a low-energy protocol in LoRa-based monitoring systems. Sensor transmission schedules are optimized to eliminate redundant data sending. Sensor transmission schedules are optimized to eliminate sending repetitive data. The network protocols are changed according to environmental temperature conditions and sensor activity.

Yang et al. [25] proposed an adaptive brightness control model that passes the controls through IoT technology. Daylight intensity, motion presence, and user preferences are recorded for each lighting zone. The dynamic occurring for each area increases or decreases brightness depending on what is needed. Lighting energy consumption has decreased without affecting visibility or comfort.

For heating and cooling performance, Boutahri et al. [26] carried out reinforcement learning for the optimal smart thermal management systems. The model adjusts the heating and cooling schemes based on a real-time activity level and the outdoor temperature. System behavior grows as more environment–response data gets collected. This translates to reduced energy consumption while continuously ensuring thermal comfort.

To efficiently harvest ambient power, Keshavarz et al. [27] developed a nonlinear circuit architecture capable of energy collection. The circuit also ingeniously switches between receiving mode, transmitting mode, and stored mode. Ambient idle energy is harnessed and converted into a storage unit for future active use. The system ensures long-term autonomous operation with minimized energy loss.

To address the shortcomings of the associated literature, the proposed CdPCM enjoys the benefits of having a deep recurrent learning network that dynamically classifies device schedules into sustainable and power-demanding modes, thereby allowing a more accurate and adaptive power management approach in IoT-based smart controllers. CdPCM, in contrast to the previous techniques where the emphasis is frequently on the static or heuristic-based power-saving techniques, utilizes the recurrent classification and scheduling, which constantly review the device power states and task assignments, and guarantee the most efficient transition between the active, sleep, and recharge modes. This learning and dynamic system enables completion of tasks and power conservation to balance well, avoids waste of power through idle devices, and does not overload the devices.

CdPCM also solves the problem of wasted power consumption and ineffective allocation of tasks by moving input controls off the overloaded controllers to maintain active states on others, thus enhancing the overall system responsiveness and power efficiency. The low-power sustainable feature used in the module to detect the maximum time the mechanism can operate before recharge and its periodical checking mechanism are also what make the module unique compared to the corresponding literature that might not capture such fine-grained transitions of power state and adaptive time schedules based on the current state of affairs in real time.

Moreover, the experimental validation of the proposed method in a realistic IoT setting that involves various devices and controllers shows that it saves power, consumes less power, and achieves complete control at a higher rate than the current approaches. This shows that CdPCM is effective in reducing the problems of power loss and failure of tasks that have been experienced in earlier methods.

# 3. Proposed Control-Dependent Process Classification Module

The proposed control-dependent process classification module (CdPCM) is described in this section. This module is designed to improve the power-saving features of the smart controllers linked with IoT paradigms. In environments such as smart industry or smart building, where device/machine control is remotely managed, this module is efficient. Based on the controls and the power constraints, the execution and completion of the inputs are achieved. A schematic view of the proposed module in a smart environment is given in Fig. 1.

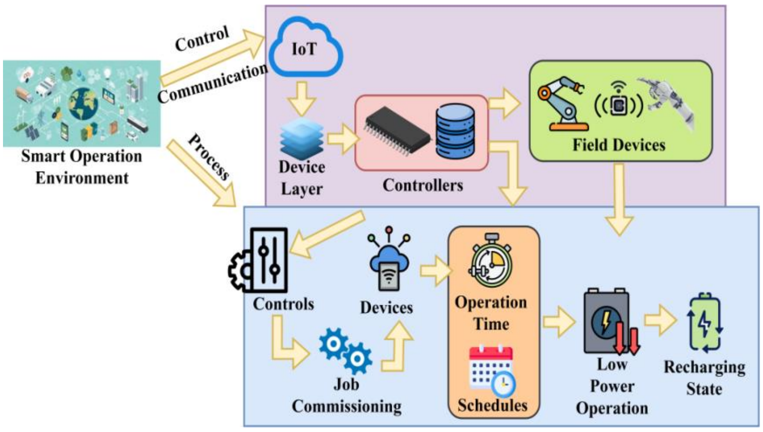

**Fig. 1:** Schematic Representation of IoT Integrated Process Module.

For ease of understanding, the IoT and process modules are depicted separately in the above Fig. 1. This isolates the process modules and the layers of the IoT. The IoT paradigm device layer deals with the controllers and allocates the input jobs to be performed. The figure identifies the impacts of the time each device operates and the time it takes to determine the power consumption and power saving periods of each device. The module makes sure that the devices go to a low-power sustainable state before reaching a recharge state and commissioning and decommissioning tasks, respectively. The allocation and power-saving scheduling in the smart controller network is based on this schematic. The device layer of the IoT paradigm is responsible for handling controllers and providing them with inputs (jobs) for completion. The operation time and schedules of a device determine its power consumption and conservation intervals. This evaluation is verified to ensure the low-power operation state is moved to the recharging state regardless of the job count. Besides, the sustainable factor of the devices (based on power) is validated by the deep recurrent learning network, accounting for their commissioning and decommissioning intervals. The variables used in the article are introduced in Table 1.

**Table 1:** Variables Used and Their Descriptions

| Variable | Description | Variable | Description |
|---|---|---|---|
| $\mathbf{N}$ | Field Devices | T | Jobs |
| $\mathbf{t_{in}}$ | Job Allocation Time | I | Interval |
| $\mathbf{t_{out}}$ | Job Completion Time | $P_c$ | Power Consumption |
| $Bold\mathbf{C}_{a}$ | Current Active | $C_s$ | Current Sleep |
| $Bold\mathbf{C}_{i}$ | Current Idle | V | Input Power Voltage |
| $\mathbf{P_S}$ | Power Saving | $T_c$ | Completed Task |
| $\mathbf{T_f}$ | Failed Task | C | Classification |
| $\mathbf{S_s}$ | Sustainable Schedule based on Power Saving | $S_c$ | Sustainable Schedule based on Power Consumption |
| $\mathbf{P_l}$ | Power Loss | | |

## 3.1. Power consumption and saving computation

A smart controller in the smart environment interfaces with N field devices to allocate T Jobs initiated at the time $t_{in}$. The total jobs processed in an interval I is $(N \times T)$ where $I = (t_{in} - t_{out})$ where $t_{out}$ Is the job completion time? The power consumption $(P_c)$ of the devices for $(N \times T)$ It is computed in the following equations.

$$P_c = \int E(t_{in})dI \tag{1a}$$

Where

$$E\,t_{in} = V * (C_a + C_s + C_i) \tag{1b}$$

Such that,

$$V * C_a * (t_{out} - t_{in}) = (t_{out} - t_{in}) * V \times C_i - V \times C_i \tag{1c}$$

$$\Rightarrow (V * C_a * I) = I * V \times C_i - V * C_i = C_a I = (I - 1)C_i = C_a I = I - 1 \text{ and } C_a = 1 - \tfrac{1}{I} \tag{1d}$$

In the above set of equations, the variables $C_a C_s, aC_i$ represent the current active, sleep, and idle states of the. The variable V, denotes the voltage of the input power, such that the condition $C_a = \left(-\frac{1}{I}\right)$ is the satisfying case for maximum utilization. This refers to the possible criteria satisfying the power need ($\times$ T) tasks in I; this ensures maximum switch over from active to idle or sleep state before a low power state is entered. The power consumption for a cumulative interval $N \in C_a$ or$)$, $C_s$ othethanha $C_i$. Advise in $C_i$ is said to be in a low-power state, whereas the other case of $C_s$ indicates that the devices need energy backup or recharge. Based on the above consumption, the power saving estimation is given in equation (2).

$$P_s = \frac{P_c - \left[V \times (C_a + C_s) \times \left(1 - \frac{t_{in}}{I}\right)\right]}{P_c} \tag{2}$$

In equation (2), the power-saving output is defined considering the active and sleep states of the devices. Based on the different power saving, the consecutive schedules for T Are performed. The classification for schedule assignment and power allocation for the devices is modeled by the deep recurrent learning network. In the $P_c$ and $P_s$, the T Allocation and completion are not the same. Therefore, the prompt T Allocation and completion are described below.

### 3.2. T allocation and completion in $P_c$ and $P_s$

This allocation process relies on the power allocated and the power consumed in I. However, post the completion of IThe allocation and completed inputs are known. The first allocation is $\sum I$ such that $\frac{T}{N}$ Is the distribution even? However, based on T length and N's $P_s$ and $P_c$The completion is decided. Besides, the task/job failures due to $P_c =$ maximum (i.e.) $(P - P_c) = 0$ is computed in $(t_{out} - t_{in})$ Interval. The completed $(T_c)$ and failed $(T_f)$ Tasks are estimated using the equations below.

$$T_c = \sum_{i=1}^{I} T_i + \left(\frac{t_{in}}{I} * \frac{t_{out}}{I}\right), \text{where } t_{out} - t_{in} > I, \text{then } T_c \neq \sum T_i \tag{2a}$$

$$T_f = \sum_{i=1}^{I} T_i - T_i * \frac{(t_{out} - t_{in})}{I}, \text{where } t_{out} > I \tag{2b}$$

In the above two equations, the task is completed and fails based on I Being computed. This computation decides the $P_c$ requirement or deficiency based on multiple I, such that the need for recharging and $P_s$ for the $T_c$ Improvement is decided. The odd case of $T_f \geq T_c$ Identifies the maximum devices that are cut off from operations due to power failures. Therefore, if a device moves from active to sleep (or) active to idle, its sustainability needs to be estimated. The power requiring I It is identified using the power depicted in Fig. 2.

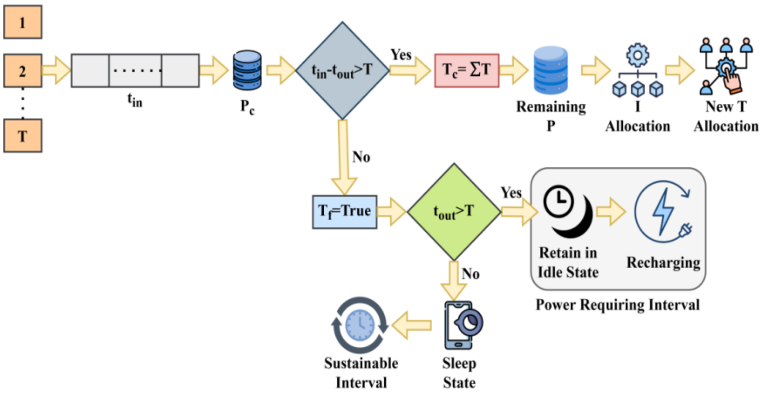

**Fig. 2:** Power Requiring T Identification Process.

The power-requiring interval is identified by categorizing two different conditions: $(t_{in} - t_{out} > T)$ is false and $(t_{out} > T)$ Is true. If these conditions are validated, then recharging from the idle state is performed. In the other cases, the remaining P (i.e.) $(P - P_c)$ is used to decide the I and T Allocation. Besides, the $t_{out} \leq T$ It is used to retain the sleep state that is actual until T Is allocated. If the devices are in a sleep state, then the change of state to consume $C_a$ is prompted only if the new T fits the current I (i.e., the new task is to be completed before $t_{out}$. If this is feasible, then the $P_s$ And the sustainability period is identified as $T_c - t_{out}$ where $T_c > t_{in}$ and $T_c$ Denotes the completion time of a T assigned to N (Fig. 2) It classifies the power states of a device according to two conditions, namely, true or false of a given condition. In case the condition has been proved to be true, the device will move out of an idle state into a recharging state. Otherwise, all the remaining power is consumed to determine the task allocation and scheduling. The process also controls the retention of the sleep state, where a device can be left in sleep until a new task can be scheduled within the current sustainable operation period. When the new activity can be accomplished by the end of the sustainable period, the device also switches the state, and the sustainable period is also modified according to the time that the given task has been accomplished. This identification aids in the dynamic control of power consumption by determining when the devices are to resume their recharging after going into the low-power states, or not, to ensure the highest efficiency and completion of tasks. Following the above identification, the power saving through a sustainable interval of the N It is computed under different constraints analyzed in the following section. This makes use of a distinguishable deep recurrent learning model to maximize savings decisions, reducing the loss.

### 3.3. Learning model for $P_s$ and loss reduction

The deep recurrent learning network model is employed in this proposed method for identifying losses and thereby providing sustainable developments on $P_s$. This learning network intake T as $P_c$ as inputs wherein the loss interval $(I_l)$ It is identified from the intermediate layers of the network. This network contains one input layer, one hidden layer, and one output layer for conditional assessment. The output is the classification of T at the end of $t_{out}$ such that $P_s$ and $P_c$ Decisions are provided for the available N for handling upcoming T. Thus, the network is trained using the current decision output for identifying the low-power sustainable schedules. Based on the decision, the $P_s$ and $P_c$ Schedules are differentiated to maximize completion with fewer losses. The classification $(\complement)$ Is defined in equation (3) series.

$$C=\begin{Bmatrix} S_s, \text{if } t_{in}-t_{out}=I \text{ and } (P-P_c)>\frac{P_s}{2}\Rightarrow P_s \text{ based decision} \\ S_c, \text{if } t_{out}>I \text{ and } (P-P_s)\leq\frac{P_s}{2}\Rightarrow P_c \text{ based decision} \end{Bmatrix} \tag{3a}$$

Where

$$S_s=\frac{\Sigma(t_{in}-T)+\Sigma(T-t_{out})}{(t_{in}-t_{out})}\times\nabla, \nabla=\frac{(t_{in}-t_{out})}{I}\times\left(C_a\times\frac{T}{N}\right) \tag{3b}$$

And

$$S_c=\sum\frac{t_{in}-t_{out}}{I}*(1-\nabla)\forall(P-P_c)=0 \tag{3c}$$

In equations (3a), (3b), and (3c), the variables $S_s$ and $S_c$ References the sustainable schedule based on $P_s$ and $P_c$. These schedules are identified under a derivative factor $\nabla$ that alters the assignment of the next T in $(I+1)$. In this derivation, the need for charging and T allocation is classified under $S_s$ and $S_c$ that is balanced with the actual interval I. Before balancing the equations based on I, the deep recurrent learning network for classification is depicted in Fig. 3 below.

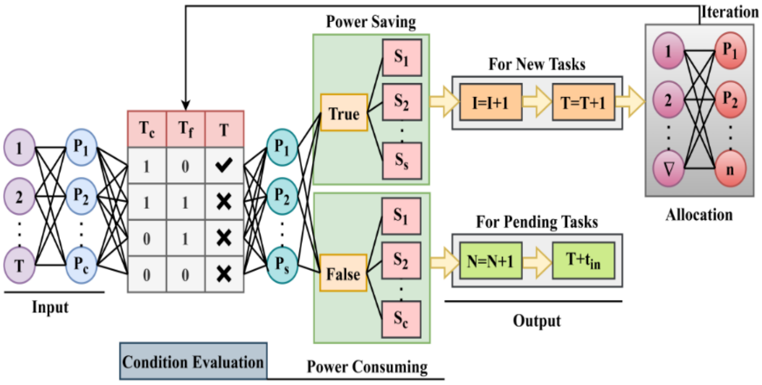

**Fig. 3:** Learning Network for C Process.

The learning network is diagrammatically presented in the above Fig. 3. The description stepwise is as follows:

Step 1: Input Layer: The network is fed with combined inputs that reflect alterations in the power status of the devices and the assignment of tasks. These inputs depict the present state of the operations of the devices and the schedule requirements.

Step 2: Hidden Layer: The network then takes the inputs and processes them by the use of a hidden layer, where repetitive connections are used to make the model remember the past. This layer filters the input data, identifies time-based dependency, which is necessary to classify dynamically.

Step 3: Output Layer: The output layer is used to distinguish the device schedules into two categories:

- Sustainable schedules: Marking low-power operation conditions, in which the devices can remain without a recharge.
- Power-demanding schedules: Specifying when more power is used that needs to be replenished or task rescheduling.

Step 4: Recurrent Refinement: That network is repeatedly updated in its classification through constantly re-evaluating the availability of devices, the state of their power, and the tasks they are given. The repeatability of this process enables the model to adjust to changing conditions and maximize the scheduling decisions.

Step 5: Decision Integration: According to the classification, the network has the power to balance the need to fulfill the goals of tasks and power-saving approaches by:

- Enabling devices to be in a sustainable low-power mode where feasible.
- Urgent charging of the rechargeable device or redistribution of tasks with the identification of power-consuming conditions.

Step 6: Recurrence Structure The recurrent nature of the network will provide constant checks and balances regarding the verification of power saving schedules and the completion rates of tasks, as well as power wastage is to be minimized. The T is mapped to $P_c$ is the actual input that combines the changes of $T_c$ and $T_f$ to identify the actual T. This T is the refinement from the input that is categorized under $P_s$ only. For the $P_s=$ true cases, $S_1$ to $S_s$ is the sustainable schedule for the $P_s=$ false cases, the $S_1$ to $S_c$ is the power demanding schedules. Therefore, for the first classification I and T are incremented; for the latter case, the device availability is checked along the $t_{in}$. This refers to the accommodation of the pending task due to a power failure or an idle state. The changes in new $I, T$ s identified by $\nabla$ and the new N assigned under $P_s$ classification to maximize the detection. This detected $(\nabla, N)$ allocation is used to update the T as the $(T_c > T_f)$ chances are high. The process is recurrent until maximum classification is reached. From this learning model, the unsatisfactory condition of $P_s=$ true has possibilities of power losses. This loss $(P_l)$ is estimated using equation (4).

$$P_l=P_c+P\left(1-\nabla*\frac{T}{N}\right), \text{where } I<t_{out} \tag{4}$$

In the above equation, the power loss computation is provided, which is used to decide the state of charging. Based on the power loss computed, the recharging state is recommended through which the device is moved from sleep to idle. In the meantime, the device overloading for the persisting task is suspended in $I_l$ where the maximum allocation is halted. This prevents the device from overloading and $P_l$ for various $t_{in}$ such that the $S_s$ from the T is the sustainable period of the devices for which $(P-P_c)>\frac{P_s}{2}$. Thus, the precise sustainable period in the low-power operation state is defined as follows:

$$\forall\, C \text{ is categorized as } S_s,\ S_s=T_c \text{ where the completion is higher than the loss} \tag{5a}$$

And

$$\frac{t_{in}-t_{out}}{I} = \frac{t_{in}-T+T-t_{out}}{t_{in}-t_{out}} \Rightarrow T = 0 \text{ is the pending after T} \quad (5b)$$

$$\forall C \in S_c, T = 1 - \left(C_a * \frac{T}{N}\right) \text{, where the } \nabla \text{ is the completion deciding factor} \quad (5c)$$

From the above equations, the sustainable period from (5b) is for T (i.e.) $(t_{out} - t_{in})$ and $P_s$ is the same as in equation (2). Besides, the minimum count of $I_l$ is the remaining sustainable period for handling new T. This is different in the equation (5c), where ∇ is the difference factor between $T_f$ and $T_c$. Therefore, the $P_c$ demands are high, and thus, the need for additional P is required. The T is disconnected from the N until the $I_l$ period to ensure the completion of jobs without $P_l$ and task failures. Therefore, the $S_s$ and $S_c$ are organized together in proper T to ensure $P_s$ is high through the learning network. Based on the number of learning iterations, a single N is recurrently verified for its low-power operation states. These verifications prevent on N from entering the idle state other than the sleep state or inactive states for any T.

## 4. Results and Discussion

The proposed model is analyzed using Contiki Cooja-based simulation to verify its output and performance. The network is modeled with 60 edge devices capable of operating between 30s and 450s for a maximum of 50 requests. In the authentication process, elliptic curve cryptography with a 160-bit key is used. The maximum validity time is 240s, and therefore, for a complete 450s interval, requires 2 $K_{re-gen}$ instances. The revocation time is between 3s and 5s for any number of $D_{auth} = 0$ users. Using a single cloud server device, 200 users are connected to serve the request with authentication.

### 4.1. Hyperparameter analysis

The hyperparameters ∇ and C are the considerable factors in this proposed module to maximize the efficiency of IoT-based smart controller operation. The $T_f$ and $T_c$ are the fundamental targets of these parameters to maximize the efficiency of the N. Therefore, the analysis of these two parameters is presented in Figs. 4 and 5.

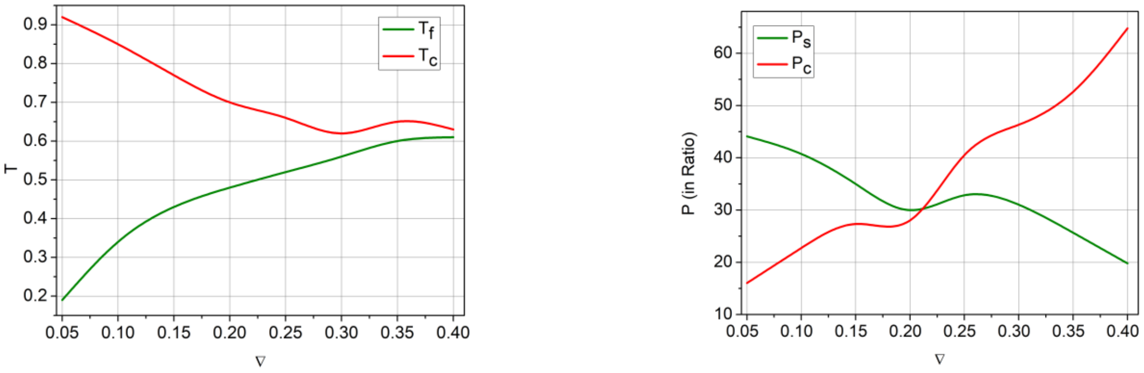

**Fig. 4:** ∇ Based T and P Analysis.

Fig. 4 shows the assessment of the hyperparameter a, which affects the rate of change in the classification of the task and power-saving performance in the suggested CdPCM. As indicated in the figure, at a the value of power saving and task failure rate will be an equal value, thus, loss of efficiency with overload. It is identified in the analysis that this limit is kept to avoid overload conditions and guarantee optimum power saving and classification of task performance. The figure also depicts the success of trials and increment of repetitions to confirm any changes, with an important consideration on the matter to save power and complete tasks. The rate of change of ∇ is considered up to 0.4, as if it crosses 0.4, then $T_f$ and $T_c$ would be equal. Therefore, any ∇> 0.4 reduces the efficiency through $P_l$ and $P_c$ due to N overload. To prevent this, the C is induced that targets ∇ reduction by defining tasks for $S_s$ and $S_c$ classes. Based on the number of successful trials, the $T_c > T_f$ is the target for power saving and failure reduction. Besides, the iterations on verifying the changes in $\left[(P - P_c) < \frac{P_s}{2}\right]$ is the consecutive condition for achieving fair $T_c$ and $P_s$ in this module. In Fig. 5, the ∇ variant analysis is presented for C.

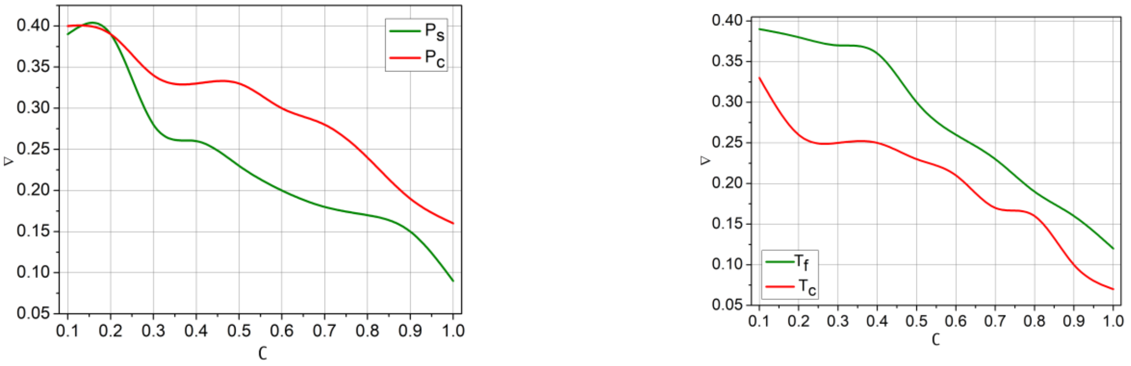

**Fig. 5:** ∇ Analysis for C Variant.

Fig. 5 evaluates how the parameter of variant, b, affects the recognition of the power-saving condition and task failure condition in the module. The figure shows the fact that b has a significant influence on the classification accuracy of sustainable and power-demanding schedules. When increases, the number of task failures increases, the system increases power saving, and minimizes failures through allocating and utilizing tasks. The interpretation will demonstrate the impact of b on the learning steps and timing activities to make sure that the amount of overloaded tasks is minimized and optimum completion of tasks is achieved. The role of $\mathrm{C}$ in detecting $\mathrm{S_s}$ and $\mathrm{S_c}$ is high in this proposed module based on $\mathrm{P_s}$. If the $\mathrm{P_c}$ results in $\mathrm{P_l}$, then $\mathrm{S_s}$ to $\mathrm{S_c}$ is adopted to maximize $\mathrm{T_c}$ and therefore the $\nabla$ is reduced. Besides, the N allocation and I utilization are definite across multiple input T such that $\mathrm{P_s}$ becomes mandatory. Therefore, the learning is defined for overloaded T for the available N such which $\nabla$ is reduced. This reduction is defined from the $\frac{\mathrm{P_s}}{2} > \left(\frac{\mathrm{P-P_c}}{2}\right)$ and $(\mathrm{T_f} < \mathrm{T_c})$ cases to ensure the iterations are optimal to maximize the task completion. Thus, the $\nabla$ occurrences are limited to the maximum $\mathrm{C}$ for T allocation in $\mathrm{S_c}$ and $\mathrm{S_s}$ schedule (Fig. 5).

### 4.2. Comparative analysis

The experimental analysis of the proposed module is performed using the Contiki Cooja simulator. A wireless network operating within a small section of a smart industry is the environment considered, with 50 field devices linked to 6 controllers. The devices are assigned 5 tasks/interval based on different lengths. In these interval allocations, 20 min to 240 min is considered to complete the tasks. The 6 controllers generate 10-12 control inputs for task execution and completion. Using this setup, the proposed module is analyzed comparatively with certain metrics. The metrics, such as power consumption, power saving ratio, control completion, and idle power loss, are compared with the existing DA-SEMS [19], CPSO-PID [18], RL-EE [26], and PEMFC-PEMEC [20] methods and are discussed below.

### 4.3. Power consumption

Fig. 6 compares the power draw of the proposed CdPCM with the current techniques (DA-SEMS, CPSO-PID, RL-EE, and PEMFC-PEMEC) with different numbers of field devices and different task periods. The figure shows that the amount of power that is saved by CdPCM when devices are switched to the idle mode or sleep is always lower, and then devices are recharged, which helps reduce power consumption. This transformation lowers the operational power consumption and does not compromise on tasks. It is also an indicator of the capability of the module to assign tasks in an efficient manner, leading to drastic reductions in total power consumption relative to other techniques.

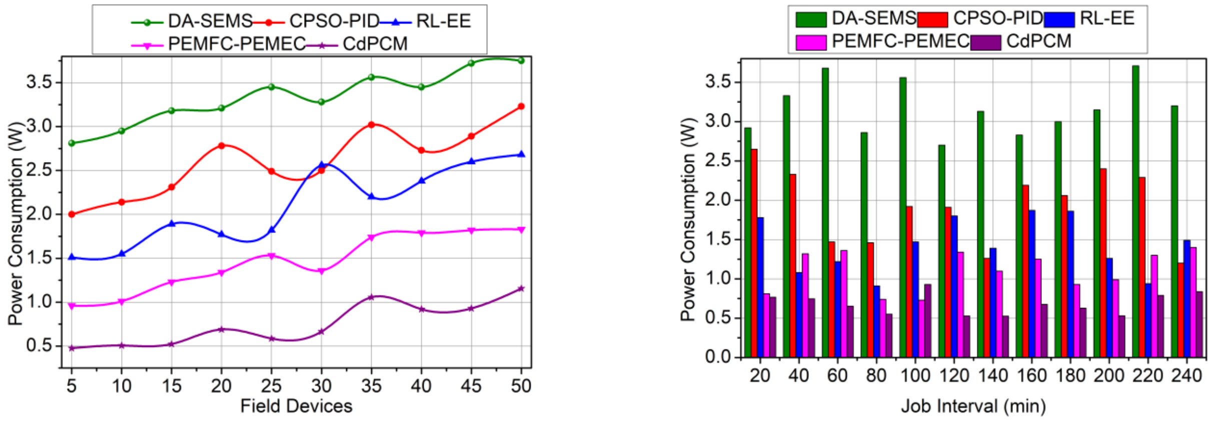

**Fig. 6:** Power Consumption.

The comparative analysis of $\mathrm{P_c}$ is presented in Fig. 6 for the varying field devices and I. The proposed module moves the devices from the active to the idle state if $(\mathrm{P} - \mathrm{P_c}) > \frac{\mathrm{P_s}}{2}$ is identified. This state change reduces the $\mathrm{P_c}$ of N (active), whereas the low-power operations are still pursued. Before it reaches the idle state, a recharging recommendation is provided to ensure $\mathrm{P_l}$ is less. Depending on the available N, the $\mathrm{P_c}$ and allocation are significant to maximize the $\mathrm{T_c}$. Therefore, the need for $\nabla$ is to reduce the target through $\mathrm{C}$ such that the iterations are repeated until the first condition is satisfied in any I. In the varying sequence of tasks and intervals for N devices, the $\mathrm{P_c}$ is comparatively less in this proposed module.

### 4.4. Power saving ratio

Fig. 7 indicates the power saving ratio of CdPCM compared to other available techniques in different device and task interval conditions. The figure points out that CdPCM is able to achieve a better power saving ratio because it is able to classify device schedules and transition between the active, sleep, and idle states. The task allocation and reduction of unnecessary active periods, which is achieved through the deep recurrent learning network, is manifested by the excellent power saving performance of CdPCM.

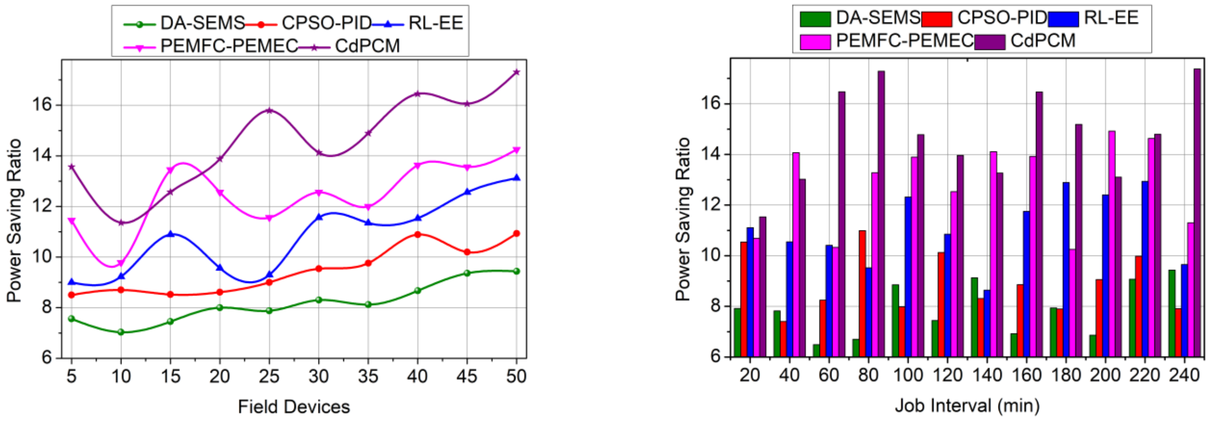

**Fig. 7:** Power Saving Ratio.

The power saving ratio of the proposed module is high (Fig.7) compared to the other methods. The $P_s$ constraints are defined based on $\nabla$ and Ϲ estimated after a single T completion. This validates the $(P - P_c)$ of the current N and $P_s$ of the previous I to ensure the device is not moving to the idle state. The learning network is responsible for categorizing $S_s$ and $S_c$ such that the $P_s$ target allocation is made for the existing T. Besides, the $P_l$ in the alternate allocation intervals as identified based on $\nabla$ to reduce its occurrences. Depending on the decisions of the deep recurrent learning, the $P_l$ and $P_s$ chances are decided. To maximize the allocations, the sleep-to-active state migrations are endorsed, where the $P_l$ is less and therefore $P_s$ is high.

### 4.5. Control completion rate

Fig. 8 involves a comparison of the control completion rate of CdPCM in relation to other methods varying in the number of devices and task intervals. The value means that CdPCM has a better completion rate since it balances the task allocation with power power-saving schedule with its learning-based classification. The module minimizes task failures and makes completion of control inputs more reliable, which represents enhanced responsiveness and efficiency of systems.

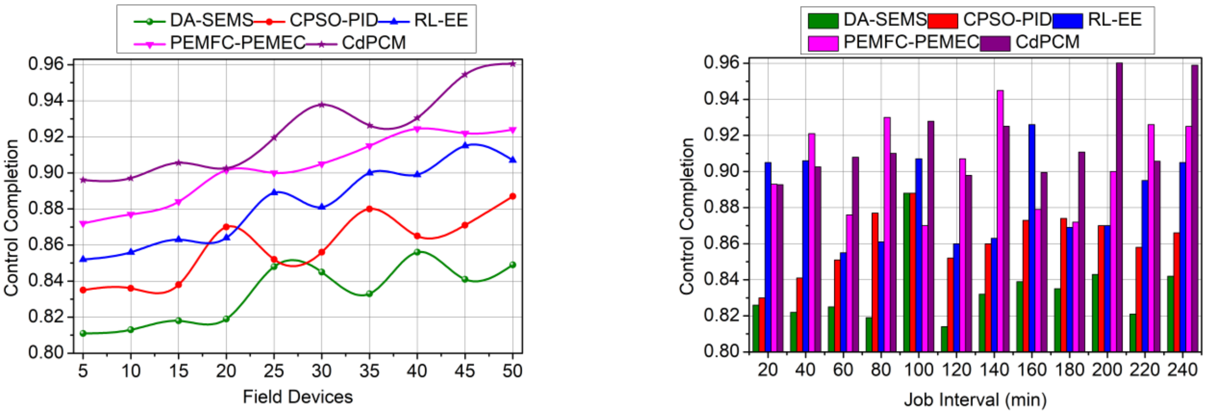

**Fig. 8:** Control Completion Rate.

The control completion rate (i.e.) $T_c$ of the proposed module is high compared to the other methods for the varying N and I (Fig. 8). The $(T_f < T_c)$ is the target condition for maximizing this completion. The $S_s$ and $S_c$ are the classified based on $P_c$ and $\nabla$ computed at the end of I. Therefore, the learning network is used to perform allocations that sustain $P_s$ or requires $P_c$. The learning decisions using Ϲ improve the chances of $T_c$ for any range of active devices. Based on the previous $t_{in}$ and $I_l$, the consecutive assignment of T to maximize Ϲ is put forward. Therefore, the number of iterations on $P_s$ and $P_c$ are independent to find the chances of $T_c$ in I and $(I + I_l)$. This improves the control task completion rate irrespective of the N with $C_a$ and N with $C_s$.
Power Loss

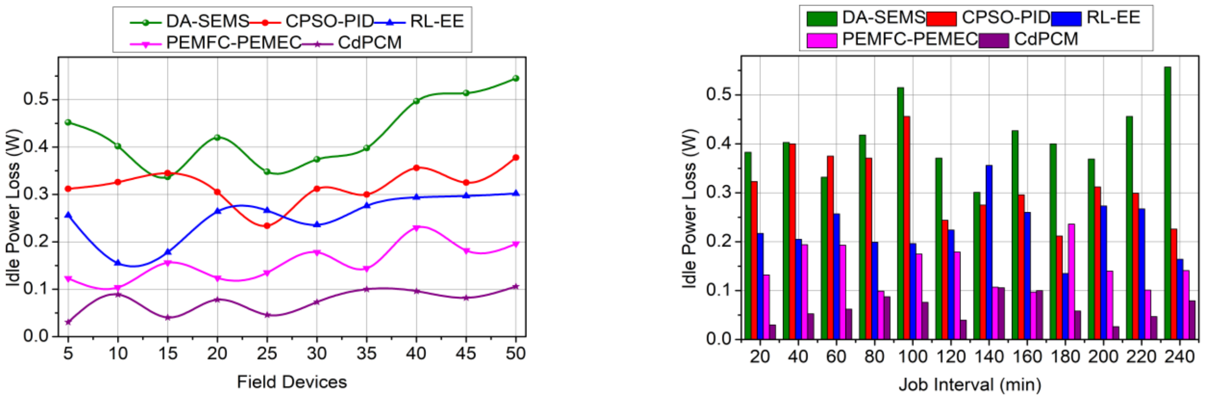

**Fig. 9:** Power Loss.

The $P_l$ for the proposed module is less compared to the existing methods, as presented in Fig. 9. The chances of active or sleep to idle state movement of a N is confined by distinguishing its $P_s$ and $P_c$ based classification. The C based classification allocates $S_s$ or $S_c$ based intervals for accepting new T and allocating I. Based on the recurrent learning network's decisions on $P_s$ and $P_c$, the above improvements are proceeding. Besides, the changes in $P_s$ based on conditions are used to validate the $T_c$ over $T_f$ and the $P_l$ reduction is initiated. In this case, the $S_c$ decides the $I_l$ before it is high, such that the device is recharged in $I_l$. Therefore, the chances of $P_l$ violating the $(P - P_c)$ is reduced to ensure N is active across various inputs. The summary of the above comparative analysis is presented in the following tables.

**Table 2:** Comparative Analysis Summary for Field Devices

| Metrics | DA-SEMS | CPSO-PID | RL-EE | PEMFC-PEMEC | CdPCM |
|---|---|---|---|---|---|
| Power Consumption (W) | 3.75 | 3.23 | 2.68 | 1.83 | 1.156 |
| Power Saving Ratio | 9.44 | 10.93 | 13.12 | 14.25 | 17.311 |
| Control Completion | 0.849 | 0.887 | 0.907 | 0.924 | 0.9605 |
| Idle Power Loss (W) | 0.545 | 0.378 | 0.302 | 0.196 | 0.106 |

The proposed module improves power saving by 11.58% and control completion by 10.77%. It reduces the power consumption by 11.22% and the idle power loss by 10.86%.

**Table 3:** Comparative Analysis Summary for Job Interval

| Metrics | DA-SEMS | CPSO-PID | RL-EE | PEMFC-PEMEC | CdPCM |
|---|---|---|---|---|---|
| Power Consumption (W) | 3.2 | 1.2 | 1.49 | 1.4 | 0.837 |
| Power Saving Ratio | 9.43 | 7.91 | 9.65 | 11.3 | 17.378 |
| Control Completion | 0.842 | 0.866 | 0.905 | 0.925 | 0.9589 |
| Idle Power Loss (W) | 0.557 | 0.226 | 0.164 | 0.141 | 0.079 |

The proposed module improves power saving by 11.22% and control completion by 10.51%. It reduces the power consumption by 11.03% and the idle power loss by 10.67%.

### 4.6. Practical application and insight discussion

The practical problems of developing and implementing the offered CdPCM are mostly associated with the difficulties of scaling and computational complexity. In a real-time case, the module should be able to distribute dynamic tasks and power between various devices and controllers, and this can be complicated further by the fact that the network being served may be growing larger. A deep recurrent learning network has to verify device states and power conditions in each control input continuously, which constitutes substantial computation overhead. This continuous calculation can bring a delay and can overload the embedded controllers, which have resource constraints, which can constrain responsiveness and efficiency. Also, it is hard to achieve a compromise between task accomplishment, power-saving schedules, and device recharging decisions because of complex iterative computations. They can partially counter the desired power savings and have an impact on the applicability of the modules to dense or large-scale IoT applications where real-time performance is important.

The power saving and computational overhead trade-off proposed in the CdPCM is the trade-off between having efficient power management and the processing requirements of the deep recurrent learning network. Although the module will enhance power saving by dynamically classifying device schedules and regulating power states, such continuous verification and classification need to consume a great amount of computational resources. The deep recurrent learning network needs to continuously evaluate the states of devices, power situations, and task assignments in real time, which adds computational load that will raise latency and power consumption on embedded controllers. This overhead can overload the devices that are resource-constrained, and can cause them to be less responsive and partially counteract the power savings accrued. Thus, although power saving is improved by more advanced scheduling and classification, it is achieved at the price of more computational complexity, and it is a trade-off in which maximizing power efficiency must be offset by processing power and energy consumption of the controllers themselves.

The Control-dependent Process Classification Module can be tailored to the various IoT applications by utilizing its dynamic and learning-based architecture that continuously categorizes device schedules and power conditions to optimize the allocation of tasks as well as power. Its rich recurrent learning architecture allows verifying and adapting the operation modes of the devices in real-time, which makes it appropriate in the context of heterogeneous IoTs having different device types, workloads, and power limitations. With the low-power sustainable feature, CdPCM can adjust the operation intervals and charge cycles to the requirements of various applications, including the smart industries, smart buildings, or healthcare monitoring systems. It can also move input controls of controllers that are congested to maintain active states in other controllers, which improves the responsiveness of the system in a wider range of deployment situations and increases its scalability. This would be extended in the future to distributed or hierarchical control architectures, which would further allow CdPCM to effectively manage computation and power in large-scale, heterogeneous IoT networks to provide adaptability and enhanced operational performance through varied applications.

## 5. Conclusion

Secure and privacy-oriented IoT device authentication needs a multi-aspect solution that considers temporal dynamics and operational integrity. To serve this concept, this article proposed the temporal authentication model. Through the incorporation of operation time and device revocation into the authentication process, the system only allows valid and timely devices to engage in the network. Distributed federated learning enhances privacy by allowing collaborative model training without centralizing sensitive information. This strengthens user privacy and reinforces the overall security infrastructure of IoT edge environments for more secure and reliable IoT deployments. Thus, the proposed model is reliable in achieving an 11.52% high authentication rate and reducing authentication complexity by 11.82% for the maximum edge devices.

## References

[1] Bhatt, K., Agrawal, C., & Bisen, A. M. (2024). A Review on Emerging Applications of IoT and Sensor Technology for Industry 4.0. *Wireless Personal Communications*, 134(4), 2371-2389. https://doi.org/10.1007/s11277-024-11054-x.

[2] Lee, B. M. (2024). Efficient Power Control Strategies for Massive MIMO in High-Density Massive IoT Networks. *IEEE Internet of Things Journal*, 11(12), 21773-21787. https://doi.org/10.1109/JIOT.2024.3378688.

[3] Huang, S., Zhang, J., Lyu, Z., & Bai, X. (2025). Research on data-driven combined network reconfiguration and local control with smart inverter for voltage regulation problem. *Energy Reports*, 13, 2000-2012. https://doi.org/10.1016/j.egyr.2025.01.051.

[4] Oh, J., Park, K., & Hwang, Y. H. (2024). A 10-Gb/s/lane, Energy-Efficient Transceiver with Reference-Less Hybrid CDR for Mobile Display Link Interfaces. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* 33(3), 887-891. https://doi.org/10.1109/TVLSI.2024.3472073.

[5] Scanzio, S., Quarta, F., Paolini, G., Formis, G., & Cena, G. (2024). Ultra-Low Power and Green TSCH-Based WSNs With Proactive Reduction of Idle Listening. *IEEE Internet of Things Journal*, 11(17), 29076-29088. https://doi.org/10.1109/JIOT.2024.3406646.

[6] El Ghati, O., Alaoui-Fdili, O., Chahbouni, O., Alioua, N., & Bouarifi, W. (2024). Artificial intelligence-powered visual internet of things in smart cities: A comprehensive review. *Sustainable Computing: Informatics and Systems*, 43,101004. https://doi.org/10.1016/j.suscom.2024.101004.

[7] Bhoi, S. K., Chakraborty, S., Verbrugge, B., Helsen, S., Robyns, S., El Baghdadi, M., & Hegazy, O. (2024). Intelligent data-driven condition monitoring of power electronics systems using smart edge–cloud framework. *Internet of Things*, 26, 101158. https://doi.org/10.1016/j.iot.2024.101158.

[8] Hadri, S., Najib, M., Bakhouya, M., Fakhri, Y., Taifour, Z., & Gaber, J. (2025). Amismart an advanced metering infrastructure for power consumption monitoring and forecasting in smart buildings. *Discover Computing*, 28(1), 1-28. https://doi.org/10.1007/s10791-025-09640-z.

[9] Bathre, M., & Das, P. K. (2024). Design & implementation of smart power management system for self-powered wireless sensor nodes based on fuzzy logic controller using Proteus & Arduino Mega 2560 microcontroller. *Journal of Energy Storage*, 97, 112961. https://doi.org/10.1016/j.est.2024.112961.

[10] Zeng, Y., Hussein, Z. A., Chyad, M. H., Farhadi, A., Yu, J., & Rahbarimagham, H. (2025). Integrating type-2 fuzzy logic controllers with digital twin and neural networks for advanced hydropower system management. *Scientific Reports*, 15(1), 5140. https://doi.org/10.1038/s41598-025-89866-5.

[11] Andreotti, A., Caiazzo, B., Fridman, E., Petrillo, A., & Santini, S. (2024). Distributed dynamic event-triggered control for voltage recovery in islanded microgrids by using artificial delays. *IEEE Transactions on Cybernetics,* 54(7), 3890-3903. https://doi.org/10.1109/TCYB.2024.3364820.

[12] Mallikarjun, P., Thulasiraman, S. R. G., Balachandran, P. K., & Zainuri, M. A. A. M. (2025). Economic energy optimization in microgrid with PV/wind/battery integrated wireless electric vehicle battery charging system using improved Harris Hawk optimization. *Scientific Reports*, 15(1), 10028. https://doi.org/10.1038/s41598-025-94285-7

[13] Tarif, M., Homaei, M., & Mosavi, A. (2025). An Enhanced Fuzzy Routing Protocol for Energy Optimization in the Underwater Wireless Sensor Networks. *Computers, Materials & Continua*, 83(2), 1791-1820. https://doi.org/10.32604/cmc.2025.063962.

[14] Mbungu, N. T., Bansal, R. C., Naidoo, R. M., Siti, M. W., Ismail, A. A., Elnady, A., & Hamid, A. K. (2024). Performance analysis of different control models for smart demand–supply energy management system. *Journal of Energy Storage*, 90(B), 111809. https://doi.org/10.1016/j.est.2024.111809.

[15] Liu, H. H., Lin, K. H., & Wei, H. Y. (2023). NR DCP: An enhanced power saving mechanism with wake-up signal enabled DRX. *IEEE Transactions on Green Communications and Networking*, 8(1), 2-17. https://doi.org/10.1109/TGCN.2023.3312707.

[16] Nsaif, M., Kovásznai, G., Malik, A., & De Fréin, R. (2024). SM-FPLF: Link-state prediction for software-defined DCN power optimization. *IEEE Access*, 12,79496-79518. https://doi.org/10.1109/ACCESS.2024.3408672.

[17] Singh, A., Salameh, H. A. B., Ayyash, M., & Elgala, H. (2024). Efficient Power Allocation and Saving Framework for VLC-Enabled Indoor Networks With Crowded Heterogeneous Obstacles. *IEEE Sensors Journal*, *24*(9), 15491-15500. https://doi.org/10.1109/JSEN.2024.3382160.

[18] Wang, X., Wang, C., Liu, Z., Kang, Y., & Wang, Z. (2025). Electric heating control of agricultural greenhouse in winter using an embedded technology based chaotic particle swarm optimization PID controller. *International Communications in Heat and Mass Transfer*, 164 (A), 108777. https://doi.org/10.1016/j.icheatmasstransfer.2025.108777.

[19] Rao, C. K., Sahoo, S. K., & Yanine, F. F. (2024). Design and deployment of a novel decisive algorithm to enable real-time optimal load scheduling within an intelligent smart energy management system based on IoT. *Energy Reports*, 12, 579-592. https://doi.org/10.1016/j.egyr.2024.06.030.

[20] Emami, A., Chitsaz, A., & Nouri, A. (2025). IoT-Driven smart energy management with a closed PEMFC-PEMEC loop: A sustainable approach to decarbonizing flexible buildings in London. *Smart Energy*, 19, 100191. https://doi.org/10.1016/j.segy.2025.100191.

[21] Sirisumrannukul, S., Intaraumnauy, T., & Piamvilai, N. (2024). Optimal control of cooling management system for energy conservation in smart home with ANNs-PSO data analytics microservice platform. *Heliyon*, 10(6). https://doi.org/10.1016/j.heliyon.2024.e26937.

[22] Zhao, L., & Yin, L. (2025). Fully-connected layers-embedded self-attention optimizer based on quantum-inspired and fuzzy logic for smart household energy management. *Sustainable Computing: Informatics and Systems*, 47, 101151. https://doi.org/10.1016/j.suscom.2025.101151.

[23] Pandey, S. K., Goyal, P., Pandey, R. K., & Panigrahi, B. K. (2025). Energy Optimization in Consumer Lightings: An IoT-Based Adaptive Control Mode. *IEEE Transactions on Consumer Electronics*, 71(1), 2285-2296. https://doi.org/10.1109/TCE.2025.3531131

[24] Zhong, C., Nie, X., & Peng, P. (2024). Novel power conservation methods for LoRa-based infrared sensors in smart building. *IEEE Sensors Journal*, 24(9), 15311-15326. https://doi.org/10.1109/JSEN.2024.3378332.

[25] Yang, L., Ma, B., Yuan, L., & Wu, B. (2024). Effective application of iot power electronics technology and power system optimization control. *Tsinghua Science and Technology*, 29(6), 1763-1775. https://doi.org/10.26599/TST.2023.9010124.

[26] Boutahri, Y., & Tilioua, A. (2025). Reinforcement learning for HVAC control and energy efficiency in residential buildings with BOPTEST simulations and real-case validation. *Discover Computing*, 28(1), 45. https://doi.org/10.1007/s10791-025-09544-y.

[27] Keshavarz, R., Sounas, D. L., Keshavarz, S., & Shariati, N. (2024). Enabling Wireless Communications, Energy Harvesting, and Energy Saving by Using a Multimode Smart Nonlinear Circuit (MSNC). *IEEE Transactions on Instrumentation and Measurement*, 73. https://doi.org/10.1109/TIM.2024.3369141.