



Resource Distribution Performance Analysis of Industrial Applications Using Peer-Dependent Scheduling and Allocation Scheme

Sura Khalil Abd

Department of Computer Engineering,
Dijlah University College, Baghdad, Iraq
mustafamusajaber@yahoo.com

Abstract:

Industrial applications widely utilize the Internet of Things (IoT) to create a smart industry to leverage the processing task. The industrial application has Peer-to-Peer (P2P) systems planned to perform resource sharing and allocation in a distributed manner for stagnancy-less task completion. The lengthy task requires several resources to enhance the stagnancy during this process. This paper proposes Peer-dependent Scheduling and Allocation Scheme (PSAS) to address this issue. The PSAS method ensures the distributed and ordered processing of tasks in the smart industry based on their length and resource availability. In the resource allocation, sharing and deadline-based features of the resources are analyzed for task computations. The method of allocation and sharing is independent as guided by predictive learning. This predictive learning lists the possibilities of task completion and shared resource availability. The peer systems assign and process the tasks based on the predictive recommendations. The peer systems interact with the IoT platform to exploit the changes in predictive computation. This learning process is unanimous, regardless of task density and resource availability. The introduced PSAS method efficiency was explored using processing time, task processing ratio, wait time, and stagnancy factor.

Keywords: Industrial applications, Internet of Things, Peer to Peer, Resource Allocation, Smart Industry, Task Scheduling.

1. Introduction

The Industrial Internet of Things (IIoT) transforms the upcoming industrial systems by sensing devices and actuators with computing capabilities and networking. The energy demand for Industrial IoT applications is increasing [1]. The industrial application uses the scheduling procedure to control, arrange the workloads, and optimize the workload scheduling to improve production. Most industrial applications use Peer-to-Peer (P2P) connections between Industrial nodes to satisfy the energy demand and improve energy efficiency by decreasing the loss while transferring the information to improve green industrial IoT [2]. Blockchain features such as nonrepudiation and non-tampering maintain industrial information in a Peer-to-Peer network. Truly distributed peer-to-peer systems tolerate node failure in the blockchain [3]. A scalable and trust-less P2P model can overcome the lack of trust between the systems; this will distribute information securely and operate transparently. Interaction between systems in the blockchain is carried out with a pair of public and private keys [4]. Systems use their Private Key to sign and transmit the transaction to one-hop peers.



P2P in a smart building with wind generators and solar panels form a microgrid that will harvest energy produced and trade energy between systems [5].

IoT approach is applied in various applications like home automation, smart cities, manufacturing applications, etc. Real-time data from industrial IoT greatly profit multiple industries' applications [6]. Traffic-aware scheduling algorithm in Industrial IoT minimizes the time for completing the task, end-to-end delay, and improves network lifetime, etc. [7]. A centralized task manager will perform scheduling by combining routing information with resources needed for communication. Time-domain scheduling will operate tasks in a TDMA (Time Division Multiple Access) manner. In the MAC layer, centralized and decentralized task solutions are included in the state of art scheduling algorithm [8,9]. Optimal scheduling is done by reducing the network duty cycle. 6TiSCH scheduling provides high reliability for industrial controlling and monitoring applications enabled by wireless communication [10,11], and radio resources are efficiently allocated for the wireless devices. Data collected for the same task in various industries is scheduled by high-priority to low-priority applications [12-14].

2. Related works

Wang et al. [15] utilized the NOMA-based approach to allocate resources in the IIoT. The system aims to reduce delay and energy consumption by reducing the subcarrier problem. The method uses online learning to resolve the task scheduling and allocation problem. To operate time slots, an iterative algorithm is used. Compared to the baseline scheme, the proposed method reduces cost.

In [16] dense industrial IoT process uses the fuzzy-related priority aware scheduling approach to improve the battery life and reduce latency; service differentiation is used along with the proposed technique. Based on priority, the allocation takes place. Improved battery life, packet ratio, and reduced latency are compared with low latency scheduling, zero scheduling, and scheduling functions.

[17] introduced the relative execution deadline scheduling approach to perform the heterogeneous routing in IIoT. The system attains reliability by applying the mixed-criticality relative scheduling algorithm, and the system's effectiveness is evaluated using the earliest deadline first algorithm, fixed priority algorithm, and relative execution deadline algorithm.

Bhatia et al. [18] recommended a quantized approach to load schedules in a fog environment in IoT applications. Node computing index is used to estimate the fog node computation capacity. To handle real-time heterogeneous tasks QCI-Neural model is used. The system efficiency is evaluated using the experimental results with scheduling models.

Jie et al. [19] proposed game-theoretic resource allocation for an Industrial IoT environment based on the fog. The fog node provides the service to data users by providing resources from the cloud center. The resource allocation problem is modeled to maximize resource utilization and achieve Stackelberg and Nash equilibrium; three algorithms are developed. The performance of the proposed method is evaluated and compared with computing schemes.



3. Proposed Peer-dependent Scheduling and Allocation Scheme

IIoT, using a peer-to-peer system identifies the task and improves the processing rate reliably by performing a schedule. The PSAS method is developed in this work to decrease the smart industry's stagnancy and waiting time. In Fig. 1, the PSAS process flow in IIoT is presented.

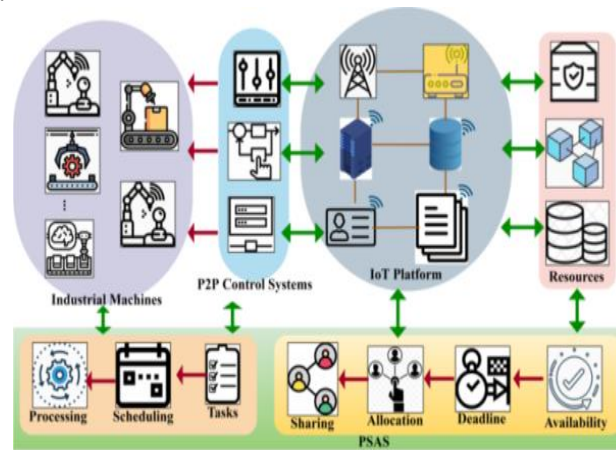


Fig 1. PSAS Process Flow

It focuses on predictive learning that deploys task scheduling and resource allocation. The task's length is identified using predictive learning, and the stagnating task is detected and overcome using the PSAS approach. In a smart industry, the resources are detected by monitoring either busy or ideal state, formulated in the following equation.

$$\rho = \left(\frac{\mu}{t_k}\right) * \prod_{x_g}^{a_b} [s_0(\vartheta) + d_a(c_e) - (q_t + e_i)] * \sum_{o_n} x_g - t_k \tag{Eq. 1}$$

In the above equation (1), the monitoring is performed for the varying resources in the smart industry to identify whether the resources are busy or ideal. Here, the resources are defined as the machines in the IIoT environment; periodic monitoring is done to allocate the tasks to the resources. The resources are represented as o_0 , the number of resources is denoted as o_n , the task is termed as t_k , the busy and ideal is referred to as s_0 and d_a . The monitoring is denoted as ρ , which deploys the preceding state of the task assigned to resources, and its computational levels are analyzed. The preceding task is termed as c_e ; it defines the queued process for the resources; in this case, the additional task is not allocated to the particular resources.

a) Task Scheduling

It is a set of protocols to be followed while assigning the task to the system that deploys the previous working state by performing a prediction process. Here, it verifies whether the task is assigned to suitable resources. The 'if and otherwise' condition is used in the following equation. It is associated with the smart industry system's state, and the computation is evaluated from the IoT environment. Two types of schedulers, long-term and short-term scheduling, deploy the resource's state of action in IIoT.



$$\nabla = \begin{cases} 1, \text{ if } \rho + \left((t_k * a_b) + \frac{s_0 - d_a}{\prod_{x_g + y' o_0}} \right) \\ 0, \text{ Otherwise} \end{cases} \quad (\text{Eq.2a})$$

The analysis is carried out by evaluating the 'if and otherwise' condition. If the condition is satisfied, the tasks are allocated; otherwise, the condition is computed. The analysis is termed as ∇ , and the periodic monitoring is performed for the varying task allocation.

The busy and ideal state is monitored periodically and allocated the task to suitable resources, and it is represented as $\frac{s_0 - d_a}{\prod_{x_g + y' o_0}}$, the density of the task is denoted as y' . The density is defined as how long the task withstands in the smart industry is measured, which is used to allocate the upcoming job by deploying the density.

b) Task Scheduling- Predictive Learning

Task scheduling is performed for the varying resources that deploy the task assigned to the system and monitor the processing. The scheduling uses the predictive learning method associated with the previous state of resources and provides the result. Thus, predictive learning uses the knowledge that affects scheduling in a planned manner in the IIoT environment. The task scheduling process is illustrated in Fig. 2. It is used to predict the pursuing task allocated with the history of resources. For this, it uses some knowledge-based retrieval.

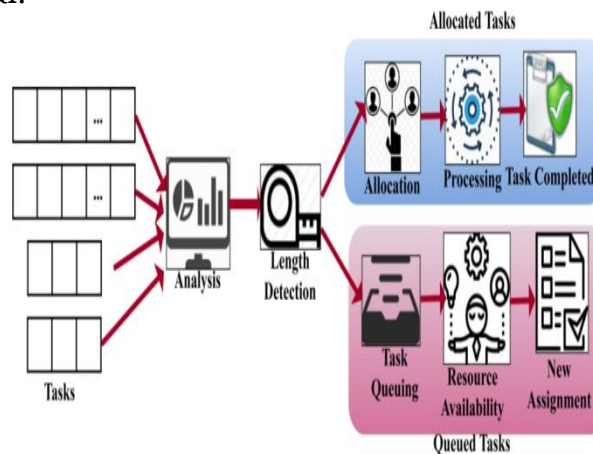


Fig 2. Task Scheduling Process

The following equation is used to identify the task with predefined knowledge.

$$h_0 = \left(\frac{s_0 + t_k}{\sum e_i \rho} \right) * \left(\mu + \frac{x_g}{o_n} \right) + ((a_b * t_k) - v_i + c_e) + \beta - u' \quad (\text{Eq.2b})$$

The task scheduling is done by utilizing the prediction model associated with the knowledge of the preceding state of the resources. The detection is carried out for the resources in the smart industry. It calculates the length of the task, and it is denoted as $\left(\mu + \frac{x_g}{o_n} \right)$. Here, the state of the system is identified, and it is denoted as h_0 , if it is busy,



the task is not allocated to the system. Here the waiting time is calculated for every task assignment, and it is represented as $\left(\frac{s_0+t_k}{\sum e_i \rho}\right)$.

Predictive learning maps the preceding task working state with the upcoming task allocated to the system; here, it measures the length and density. The allocation of the task is evaluated the available state of the resources, and it is termed as v_i , and the map with the preceding task working and is denoted as $((a_b * t_k) - v_i + c_e)$. The prediction is referred to as β , to define the pursuing state of the system reliably; by deploying this, task scheduling is done. The length and density of the task vary for every assignment. For calculating this, the below equation is equated as follows.

$$h_0(t_k) = \left(\frac{1}{o_n}\right) * \prod_{v_i}^{n_0} (\rho + q_t) * \left[(s_0 - d_a) + \left(a_b(d_a)/\beta\right)\right] - (\nabla - k_e) \quad (\text{Eq.3})$$

Identifying task length and density are defined by relating them with the preceding state of processing that deploys predictive learning. Task scheduling is estimated for varying resources, periodic monitoring is carried out, and the system's available and unavailable state is identified. The unavailable is defined as n_0 , deploying this task is provided to the ideal system, and it is represented as $\left[(s_0 - d_a) + \left(a_b(d_a)/\beta\right)\right]$.

If the length and density are defined, then the scheduling is estimated reliably, and in this, queuing is monitored; if the process is in the queue, it waits for the ideal state. By processing this, the system maintains the busy state, and no processes are queued, and it shows better performances of a task in the IIoT. Thus, the task's length and density are identified by equating the above equation to perform the scheduling. The verification is used to share the task with the other ideal resources. For this, an availability and unavailability check is performed, formulated below.

$$\partial = \prod_{o_n}^{\rho} v_i(o_0) + t_k(d_a) + a_b - c_e + \left(\frac{r'+o_0(d_a)/\vartheta}{x_g+y'}\right) \quad (\text{Eq.4a})$$

The verification phase is done by evaluating the above equation (4a) to check the availability and unavailability of resources and provide the task accordingly. If the resource is available, the task is allocated to the desired system and promptly monitors the computation, which enhances the processing time. Thus, the task is allocated with the predefined knowledge to analyze the upcoming task. The length and density are calculated to allocate the new task to the system for this previous history of processing is necessary to perform the scheduling, and it is denoted as $\left(\frac{r'+o_0(d_a)/\vartheta}{x_g+y'}\right)$.

The verification is termed as ∂ helps to analyze the accessibility and inaccessibility of the resources, and then the sharing r' is processed among the resources. Only the ideal systems are provided with the task; thus, the assignment is performed post this verification phase. In this scheduling method, predictive learning allocates the task to the smart industry's resources to decrease stagnancy. The following equation maps the preceding task working with the pursuing task and detects the stagnancy to address this methodology.

$$\delta = \sum_{o_n} (s_0 - d_a) + \nabla * \left(\frac{o_0+v_i}{r'}\right) * \prod_{\vartheta} \left(k_e + \frac{e_i}{q_t}\right) + \rho - (x_g + y') \quad (\text{Eq.4b})$$



The mapping is equated in the above equation represented as δ ; in this, it detects the busy and ideal state of the system by evaluating the predictive learning method. It is associated with the knowledge of the previous state of resources and provides the task to the available resources in IIoT. Thus the mapping is estimated by acquiring the upcoming task associated with the prediction method. It results in the sharing of the task, and it is denoted as $\nabla * \left(\frac{o_0+v_i}{r'}\right)$. Fig. 3 shows the task mapping for the devices using predictive learning.

Mapping is performed by relating the preceding data state with the pursuing state, resulting in reliable scheduling that increases the processing ratio. In this proposed work, the mapping is done for every instance to perform the efficient scheduling associated with the queued task. Here the stagnancy is not properly identified.

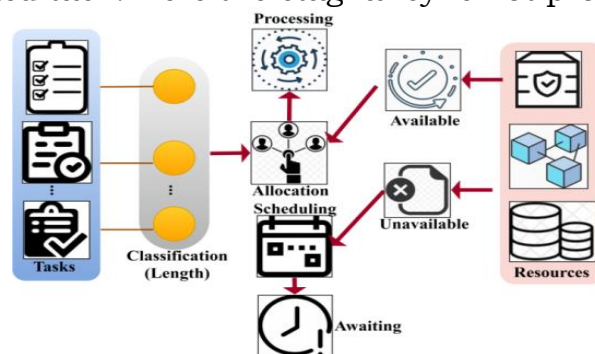


Fig 3. Task Mapping using Predictive Learning

The following equation (5) is used to detect stagnancy and verify shared resources and mapping.

$$\delta(r') = \left(t_k + \rho/q_t\right) * \sum_{k_e}(e_i + o_0) * \left(\frac{s_0+d_a}{\vartheta}\right) + (h_0 - \beta) * a_b(d_a) \tag{Eq.5}$$

The task mapping to the ideal system is estimated by scheduling; this periodic monitoring of resources is done to analyze the availability. If the resources are available, the scheduled task is given to the system and monitors the working state that deploys to detect stagnancy. Stagnancy is defined as the task is not completed on time; to address this, resource sharing is performed on the available task that is ideal or not. Here the queued process is monitored, and it is denoted as $\left(t_k + \rho/q_t\right)$.

In this equation, the identification is performed by utilizing the prediction model. This shares the ideal resources, and it is denoted as $(h_0 - \beta) * a_b(d_a)$. The task scheduling in predictive learning is used to share the resources to the ideal system and address the waiting time of the process that relies on the queue. It is formulated in the following equation.

$$\vartheta = \left(\frac{\rho+o_n}{s_0-d_a}\right) * a_b + \prod_{\delta}(c_e - u') * \left(t_k * o_0/m' * \nabla\right) + \sum r'(o_0 - a_b) - q_t \tag{Eq.6}$$

The scheduling is performed by deploying the predictive learning method associated with the system's busy or ideal state; this method uses periodic monitoring to identify stagnancy. If stagnancy is detected, the processing ratio is improved. For this analysis, the previous step of the task allocation process is necessary to map. This mapping



identifies the task allocation to the respective resources and monitors the stagnancy, and it is represented as m' .

c) Resource Allocation

The resource allocation is sharing the available resources to the ideal system and verifying whether it works properly or not, and it is monitored periodically. It manages the tasks, assigns them to the available resources, and calculates the waiting time of the resource in the queue to decrease the processing ratio. In Fig. 4, the resource allocation process is illustrated.

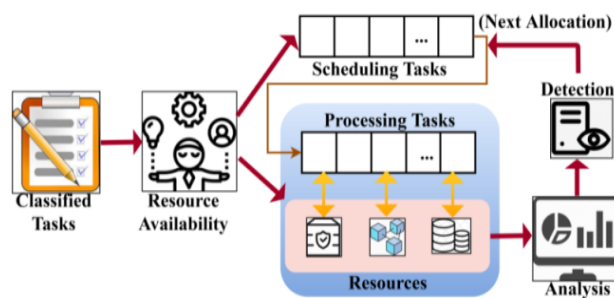


Fig 4. Resource Allocation Process

The allocation of resources is used to promptly evaluate the number of tasks to the ideal system and estimate its length and density. The resources are analyzed by equating the following equation to share the task with the system.

$$\nabla(o_0) = (r' + t_k) * \left(\frac{q_t - m'}{\rho / k_e}\right) + \prod_{s_0}(t_k - h_0) * \left(\frac{c_e - u'}{\beta}\right) + \delta \tag{Eq.7a}$$

The resource analysis is carried out to share the task to the desired system; in this process, periodic allocation monitoring is done to address the stagnancy. The queued tasks are analyzed to decrease the stagnancy, and it is denoted as $(r' + t_k) * \left(\frac{q_t - m'}{\rho / k_e}\right)$ here it deploys the length of the task. If the task length and density are calculated, task-sharing analysis is performed reliably, improving the processing ratio. Predictive learning is used to deploy the resource allocation for the varying resources in the smart industry that describes the previous knowledge.

d) Resource Allocation-Predictive Learning

In predictive learning, resource allocation is done by planning the system's availability in the IIoT and assigning the task by detecting the queued task. If the queued task is addressed, the stagnancy is decreased. in this processing, the resource allocation is given to the ideal system that completes the task on time. The analysis of task completion is estimated by mapping with the previous state of action, describing the task's length and density. The duration of the queue task leads to stagnancy. It is checked by equating the following equation that deploys the available resource relies upon or not by performing detection.



$$\mu = (o_0 + r') * \left(\frac{\nabla+m'}{a_b}\right) + \sum_{\delta} h_0(s_0) * \left(\frac{t_k/\partial}{e_i+m'}\right) + (x_g * y') - k_e \quad (\text{Eq.7b})$$

The detection is done equated in the above equation (7b). In this identification of resources are allocated, the analysis is performed, and it is represented as $(o_0 + r') * \left(\frac{\nabla+m'}{a_b}\right)$. The mapping is performed for the preceding and pursuing resources and allocates the task desirably by deploying predictive machine learning. The detection process is used to evaluate the resource availability and queued process that takes more time to complete the task that is denoted as $\left(\frac{t_k/\partial}{e_i+m'}\right)$. The following equation evaluates a task's identification and computation time to reduce the processing time.

$$h_0(k_e) = \left. \begin{aligned} & \Pi_{\delta}(\beta * q_t) + \left(\rho * \frac{m'}{c_e}\right) * (r' * o') \\ & = \left(\frac{r'+t_k/o_n}{\rho}\right) * (o' - \nabla) - (\beta - m') \end{aligned} \right\} \quad (\text{Eq.8a})$$

In the above equation, the processing time is computed by deploying the resource allocation to the ideal system; this predictive learning is used to predict the state of upcoming resources working. The detection process is evaluated for the varying resources in a P2P manner and provides a better task allocation. The preliminary step is to analyze the state of the resources and whether the process is queued; based on this, the task is assigned to the neighbor resources, termed as o' . Here the analysis is carried out for the neighboring system, performs the allocated task, and states the predictive learning.

Predictive learning maps the preceding with the pursuing task and finds whether the particular system is ready to execute the task. In this method, the processing time is decreased and shows a better processing ratio, and the monitoring of the task is evaluated by equating $\left(\frac{r'+t_k/o_n}{\rho}\right)$. Thus, the processing time is decreased by formulating the above equation (8a); the resource availability check is estimated for varying times and computed in the below equation.

$$o_0(v_i) = \left(\frac{1}{k_e(x_g+y')}\right) * q_t - \left[(h_0 * d_a) + \left(\frac{t_k/a_b}\right) \right] * (\partial - k_e) + c_e * r'(o') \quad (\text{Eq.8b})$$

The resource availability check is estimated for the number of systems in the IIoT; both the busy and ideal resources are available. Here, it verifies the system's state and provides the result that deploys the allocation process to the resources in a P2P manner. The task's length and density are identified, and sharing the resources with the neighboring system improves performance. This verification of resource availability is executed for every state of the allocation process that makes the resource allocation to the ideal system. Computing improves the processing ratio and decreases the waiting time of the process. The task is enlisted in the queue responsible for sharing the resource with the other system. This process is evaluated in the following equation (9a).

$$\rho(l_i) = (m' * h_0) + \left(\frac{r'}{o_0}\right) * o' + \left(\frac{\nabla}{\sum v_i * c_e}\right) * \sum_{m'}(\delta + \beta) - k_e \quad (\text{Eq.9a})$$



The queued process in the IIoT application is monitored by equating the above equation, and it is represented as l_i , it deploys the predictive learning approach that maps the preceding task state with the pursuing state and produces the output. It relates to sharing resources with the neighboring resources, and by assigning this task to the other system in the IIoT, the processing is done efficiently. The enlistment holds the previous state of the task to be performed and is monitored to identify stagnancy. Thus, the enlist of task allocation is performed reliably, and the P2P in IIoT processing is analyzed by deriving the following equation.

$$\nabla(o') = \prod_{qt} \left[(e_i * a_b) + \left(\frac{d_a * \mu}{\beta} \right) \right] * \delta - a_b(h_0) - r' + arg \max r'(d_a + h_0) - k_e \quad (\text{Eq.9b})$$

The identification phase does the sharing of resources on time, and it is termed as $r'(d_a + h_0) - k_e$. Thus, the P2P in IIoT is processed in the above equation.

4. Results and Discussion

This section discusses the performance of the proposed PSAS using simulation models. In this simulation, 1500 tasks of 50-500Mb length are processed using 50 processing devices/ machines. The simulation uses 80 scheduling instances with a queuing rate of 10 tasks/ instance. The performance is verified using the metrics task processing ratio, processing time, stagnancy factor, wait time, resource utilization, and task acceptance ratio. The tasks and task length are varied for analyzing the above metrics. For a comparative study, the existing MCRD [17], QA-HTS [18], and GT-RA are accounted for.

a) Task Processing Ratio

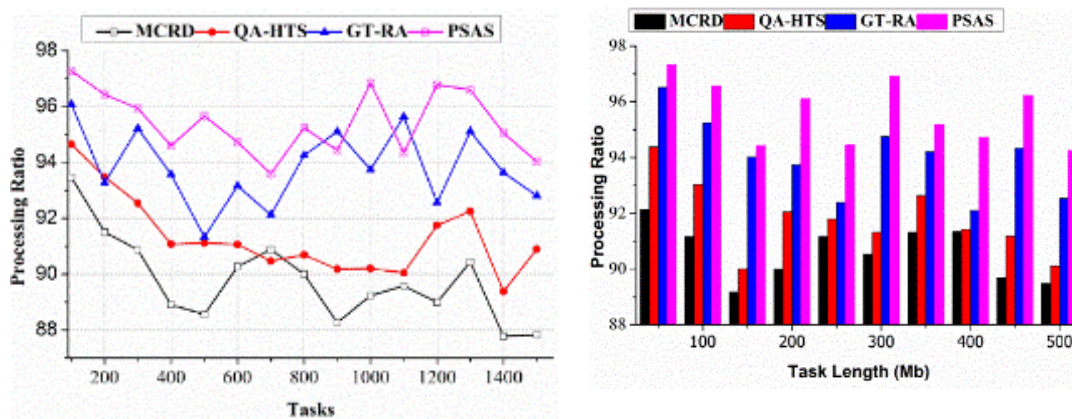


Fig 5. Task Processing Ratio for Tasks and Task Length

In Fig. 5, the task processing ratio for the proposed work increases for the varying task and task length, which is represented as $\left(\frac{\mu}{t_k} \right)$. Here, the resources in the IIoT are detected and provide the task by deploying scheduling. The prediction model analyzes the preceding information and provides the task allocation. Here, the stagnant information is detected by periodic monitoring used to evaluate the state of the



resources. In this task processing ratio, the analysis is used to evaluate the allocation process, and it is denoted as $\frac{s_0 - d_a}{\prod_{xg+y' o_0}}$. The task length and density are identified by evaluating the verification phase. In this process, task scheduling is done for the varying resources in the smart industry.

b) Processing Time

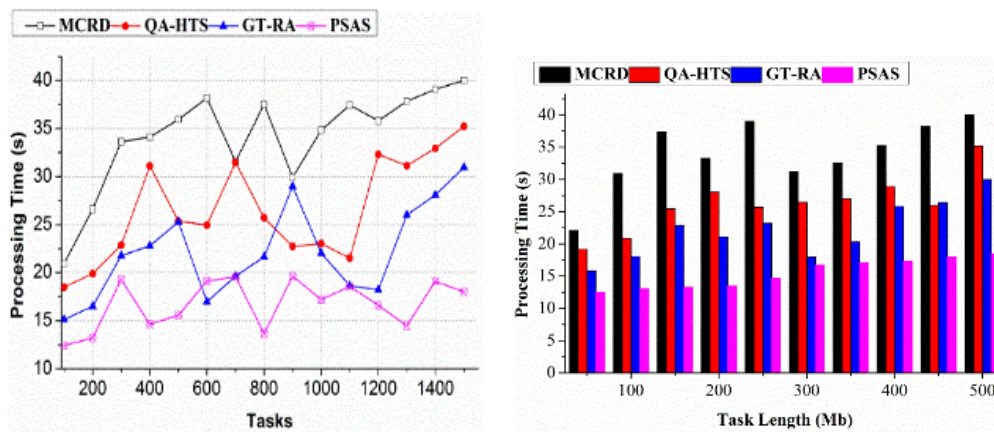


Fig 6. Processing Time for Tasks and Task Length

The processing time decreases for the varying task and task lengths, analyzing the available state of the system in the IIoT (Fig. 6). The scheduling task is used to find the stagnancy, compare it with the previously available task, and provide the result. The processing time is used to evaluate the allocated task, and it detects the length and density of the task. The busy state of the system is identified by equating $\left(\frac{s_0 + t_k}{\sum e_i \rho}\right) * (\mu + \frac{x_g}{o_n})$, here the length and density of the task are detected. The stagnancy is detected, performed by deploying the prediction phase, and mapping is carried out. In this process, the time is monitored by deploying the task's allocation to the ideal resources. By deploying this, the system performs the allocated task and provides the result on time, and it is computed as $(\nabla - k_e) * (s_0 - d_a)$. Thus, the proposed work's processing time shows a lesser value range than the other three methods.

c) Stagnancy factor

The stagnancy factor decreases for the proposed work than the other three existing methods that vary for tasks and their length. The stagnancy is monitored in the queued tasks. It determines the number of tasks being processed, represented as $v_i(o_0) + t_k(d_a) + a_b$. Here, the task is allocated to the ideal resources in the IIoT and monitors the processing time. The stagnancy is monitored by periodic monitoring of tasks and the allocation factor that deploys the mapping. In this method, task scheduling is done by deploying predictive learning and mapping the information. If the system seems to be in the Free State, it is allocated to the particular system. Here the prediction learning is



used to evaluate the length and density of the task, and it is denoted as $\prod_{\vartheta} \left(k_e + \frac{e_i}{q_t} \right) + \rho - (x_g + y')$. Thus, the stagnancy is monitored in the queued and addresses by evaluating the scheduled task's predictive learning (Fig. 7).

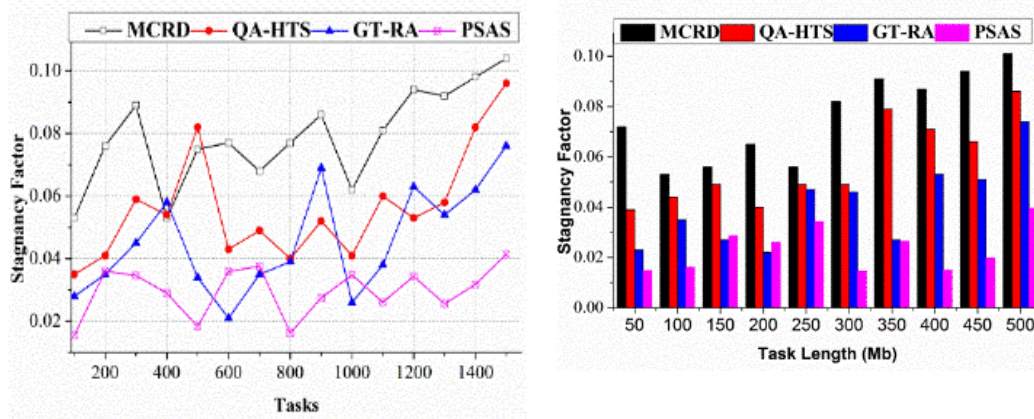


Fig 7. Stagnancy Factor for Tasks and Task Length

d) Wait Time

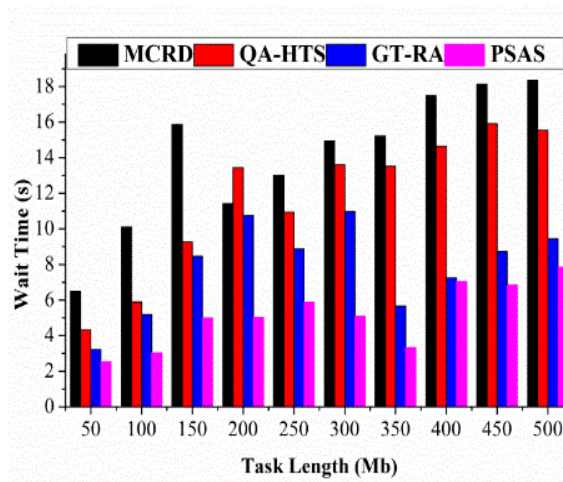


Fig 8. Wait Time for Task Length

The wait time is calculated to analyze the task length in the IIoT used to share the resources. If the system is busy, the task is allocated to the neighboring system. In this, the verification is performed to detect the system's state. The detection of the system state is done by formulating $\rho * k_e + \frac{d_a - s_0}{\beta}$, here the prediction is used to evaluate the resources. From that, it detects the length and density of the task. Only the ideal resources are allocated with the task that deploys the scheduling and allocation, and it is equated $\left(\frac{s_0 + d_a}{\vartheta} \right) + (h_0 - \beta)$. This queued task is monitored and shared with the neighboring resources by evaluating the scheduling and resource allocation (Fig. 8).



e) Resource Utilization

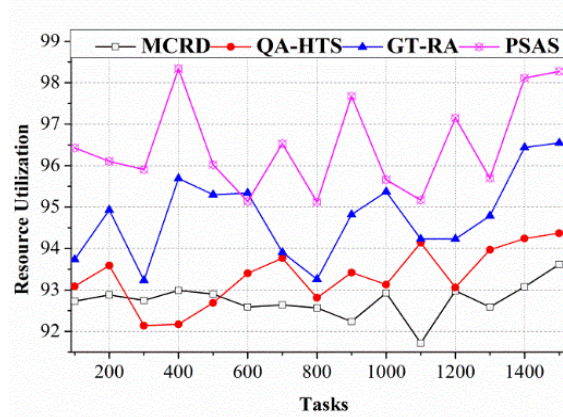


Fig 9. Resource Utilization for Tasks

The proposed work's resource utilization factor indicates a better analysis of the available system and shares the task. Task scheduling is evaluated by detecting the busy and ideal state for every instance; the task and state differ. The detection process of the state is done by equating $(t_k * o_0 / m' * \nabla)$. Here the stagnancy is analyzed. The analysis finds the task's length and density and addresses the processing time and decreases. It shows better resource utilization for the task allocation evaluated by deploying the scheduling (Fig. 9).

f) Task acceptance

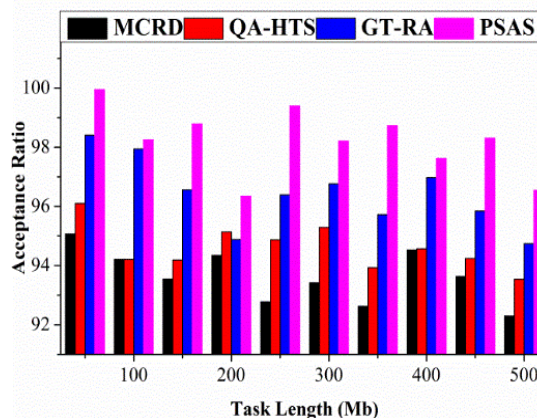


Fig 10. Acceptance Ratio for Task Length

In Fig. 10, the smart industry's task acceptance for varying tasks is deployed to identify the system's busy and ideal state. The task is assigned to systems and monitors the performance, and it is denoted as $(t_k - h_0) * (\frac{c e^{-u}}{\beta})$. The preceding state of the system is identified, and the task is allocated reliably. For the varying task, the scheduling is done, and then the resource allocation is performed, addresses the stagnancy, and detects the length and density of the task. The acceptance of the task is carried out for



the ideal system associated with the prediction phase. Here, the acceptance ratio for the proposed work increases for varying tasks that deploy the availability detection. The verification is carried out to analyze the queue process and decrease the stagnancy, and it is denoted as $\left(\frac{t_k/\partial}{e_i+m'}\right) - \rho$. The monitoring of task acceptance is used to deploy the scheduling and resource allocation phase. Here, the task acceptance ratio increases by performing integration of scheduling and resource allocation.

5. Conclusion

A peer-dependent scheduling and allocation scheme is proposed in this paper for aiding stagnancy-less task processing in peer-to-peer-assisted industrial IoT applications. The proposed scheme gains knowledge of the task length for mapping them without delay. In this process, the available resources are detected and are sequentially allocated for the first-come, first-serve tasks. The queued time-requiring tasks are assigned based on the scheduling prediction instances. For this purpose, predictive learning is employed in industrial applications to improve production; the learning process recommends allocating tasks based on completion time. The deadline and completion time of the tasks are prominent in ensuring peer-system-based task processing. The problem of stagnancy due to varying tasks and lengths is addressed using predictive learning and availability-based resource allocation. The task scheduling and resource allocation processes are concurrent in providing seamless processing with less wait time. The proposed scheme reduces the processing time and stagnancy factor and improves the task acceptance rate.

6. References

- [1]. Lee, Y., Lee, K. M., & Lee, S. H. (2020). Blockchain-based reputation management for custom manufacturing service in the peer-to-peer networking environment. *Peer-to-Peer Networking and Applications*, 13(2), 671-683.
- [2]. Yang, C., Li, X., Yu, Y., & Wang, Z. (2019). Basing Diversified Services of Complex IIoT Applications on Scalable Block Graph Platform. *IEEE Access*, 7, 22966-22975.
- [3]. Yadav, A. K., Singh, K., Amin, A. H., Almutairi, L., Alsenani, T. R., & Ahmadian, A. (2023). A comparative study on consensus mechanism with security threats and future scopes: Blockchain. *Computer Communications*, 201, 102-115.
- [4]. Munin-Doce, A., Diaz-Casas, V., Trueba, P., Ferreno-Gonzalez, S., & Vilar-Montesinos, M. (2020). Industrial Internet of Things in the production environment of a Shipyard 4.0. *INTERNATIONAL JOURNAL OF ADVANCED MANUFACTURING TECHNOLOGY*.
- [5]. Park, K. T., Lee, D., & Noh, S. D. (2020). Operation procedures of a work-center-level digital twin for sustainable and smart manufacturing. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 1-24.
- [6]. Kaur, K., Garg, S., Kaddoum, G., Ahmed, S. H., & Atiquzzaman, M. (2019). KEIDS: Kubernetes-Based Energy and Interference Driven Scheduler for Industrial IoT in Edge-Cloud Ecosystem. *IEEE Internet of Things Journal*, 7(5), 4228-4237.
- [7]. Jong, W. R., Chen, H. T., Lin, Y. H., Chen, Y. W., & Li, T. C. (2020). The multi-layered job-shop automatic scheduling system of mould manufacturing for Industry 3.5. *Computers & Industrial Engineering*, 149, 106797.



-
- [8]. Yaman, A., van der Lee, T., & Iacca, G. (2023). Online distributed evolutionary optimization of Time Division Multiple Access protocols. *Expert Systems with Applications*, 211, 118627.
- [9]. Mao, Y., Dizdar, O., Clerckx, B., Schober, R., Popovski, P., & Poor, H. V. (2022). Rate-splitting multiple access: Fundamentals, survey, and future research trends. *IEEE Communications Surveys & Tutorials*.
- [10]. Hazarika, A., & Rahmati, M. (2023). Towards an evolved immersive experience: Exploring 5G-and beyond-enabled ultra-low-latency communications for augmented and virtual reality. *Sensors*, 23(7), 3682.
- [11]. Liu, X., & Zhang, X. (2019). NOMA-based resource allocation for cluster-based cognitive industrial internet of things. *IEEE Transactions on Industrial Informatics*, 16(8), 5379-5388.
- [12]. Li, N., Xiao, M., Rasmussen, L., Hu, X., & Leung, V. (2020). On Resource Allocation of Cooperative Multiple Access Strategy in Energy-efficient Industrial Internet of Things. *IEEE Transactions on Industrial Informatics*.
- [13]. Jalili Marandi, S., Golsorkhtabaramiri, M., Hosseinzadeh, M., & Jafarali Jassbi, S. (2022). IoT based thermal aware routing protocols in wireless body area networks: Survey: IoT based thermal aware routing in WBAN. *IET Communications*, 16(15), 1753-1771.
- [14]. Sun, W., Liu, J., Yue, Y., & Jiang, Y. (2019). Social-Aware Incentive Mechanisms for D2D Resource Sharing in IIoT. *IEEE Transactions on Industrial Informatics*, 16(8), 5517-5526.
- [15]. Wang, K., Zhou, Y., Liu, Z., Shao, Z., Luo, X., & Yang, Y. (2020). Online Task Scheduling and Resource Allocation for Intelligent NOMA-Based Industrial Internet of Things. *IEEE Journal on Selected Areas in Communications*, 38(5), 803-815.
- [16]. Kharb, S., & Singhrova, A. (2019). Fuzzy based priority aware scheduling technique for dense industrial IoT networks. *Journal of Network and Computer Applications*, 125, 17-27.
- [17]. Xia, C., Jin, X., Xu, C., Wang, Y., & Zeng, P. (2020). Real-time scheduling under heterogeneous routing for industrial Internet of Things. *Computers & Electrical Engineering*, 86, 106740.
- [18]. Bhatia, M., Sood, S. K., & Kaur, S. (2020). Quantumized approach of load scheduling in fog computing environment for IoT applications. *Computing*, 1-19.
- [19]. Jie, Y., Guo, C., Choo, K. K. R., Liu, C. Z., & Li, M. (2020). Game-Theoretic Resource Allocation for Fog-Based Industrial Internet of Things Environment. *IEEE Internet of Things Journal*, 7(4), 3041-3052.