Journal of Intelligent Systems and Internet of Things
Vol. 18, No. 02, PP. 157-168, 2026
DOI: https://doi.org/10.54216/JISIoT.180212

Design and Optimization of Energy-Efficient Wireless Sensor Networks for Industrial Automation

Maha A. Hutaihit[1,*], Samir I. Badrawi[2], Haider Makki Alzaki[1] , Riyadh Khlf Ahmed[1], Marwa Falah Hasan[1]

[1]Communication Dept. Collage of Engineering, University of Diyala, Baqubah, Iraq

[2]Communication Dept. Collage of Engineering, Al Mustaqbal University, Babel, Iraq

Emails: mahaabbashutaihit@uodiyala.edu.iq; samir.badrawi@uomus.edu.iq; haider_maki@uodiyala.edu.iq; riyadh_alazawi_eng@uodiyala.edu.iq; marwaflah146@gmail.com

Abstract

To enhance the efficiency of edge-integrated Industrial IoT (IIoT) networks, this paper proposes a deep learning-based resource-scheduling framework for optimized asset booking in Wireless Sensor Networks (WSNs). The novelty of this work lies in the integration of a hybrid Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU) model, which enables intelligent allocation of computational resources based on real-time asset demand characteristics. The proposed model is evaluated using the Intel Berkeley WSN dataset and demonstrates superior performance in terms of latency reduction, execution time, and resource utilization compared to conventional approaches such as Genetic Algorithm (GA), Improved Particle Swarm Optimization (IPSO), Long Short-Term Memory (LSTM), and Bidirectional Recurrent Neural Network (BRNN). With a maximum efficiency of 99.48% and the lowest observed average delay, the model proves effective for real-time industrial automation scenarios. This research contributes to the development of scalable, energy-efficient, and responsive WSN architectures by leveraging deep learning for asset booking in edge-IoT environments.

Received: February 23, 2025 Revised: May 25, 2025 Accepted: July 12, 2025

Keywords: Cognitive industrial internet of things; Electric-field measurement system; Radio-Access Network-As-A-Service; Multi-InputMulti-Output

1. Introduction

As of late, there has been a ton of interest in wireless sensor networks (WSNs) from both industry and academia. Industrial field control, smart homes, smart factories, environmental observing, and industrial field checking are some of the many communicated observation and control areas that make extensive use of WSNs [1].WSNs are often made up of several sensor hubs that perform a variety of tasks, including data handling, transmission, and gathering. A WSN clearly offers advantages over conventional wired systems with regards to cost, adaptability, and ease [2].

Batteries in industrial settings and these batteries' constrained capacity often fuel wSNs and dreary replacement prerequisites have turned into the primary barriers to WSN adoption [3]. In WSN applications, lifetime power utilization management is crucial [4].The goal of creating energy-assortment advances is to increase the valuable lifetime of WSNs by using various procedures for energy harvesting [5]. Highlight a vibration energy-controlled WSN demonstration testbed. Vibration energy is planned to be harvested by an electromagnetic harvester [6].

This research thoroughly details its methodology, utilizing the Intel Berkeley WSN dataset and various performance metrics for evaluation. It then presents the concatenated deep learning model, explaining the mathematical underpinnings of its CNN and GRU components. The study concludes with a comprehensive performance evaluation, demonstrating the proposed model's superior efficiency, minimal delay, and faster response and execution times compared to existing techniques.

The main contributions of this research are:

- Proposing a deep learning-based resource-scheduling framework for optimized asset booking in Wireless Sensor Networks (WSNs) to enhance the efficiency of edge-integrated Industrial IoT (IIoT) networks.
- Integrating a hybrid Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU) model for intelligent allocation of computational resources based on real-time asset demand characteristics.
- Evaluating the proposed model using the Intel Berkeley WSN dataset, demonstrating superior performance in terms of latency reduction, execution time, and resource utilization compared to conventional approaches like Genetic Algorithm (GA), Improved Particle Swarm Optimization (IPSO), Long Short-Term Memory (LSTM), and Bidirectional Recurrent Neural Network (BRNN).
- Achieving a maximum efficiency of 99.48% and the lowest observed average delay, proving the model's effectiveness for real-time industrial automation scenarios.
- Contributing to the development of scalable, energy-efficient, and responsive WSN architectures by leveraging deep learning for asset booking in edge-IoT environments.

The remainder of the study details the research methodology, outlining the dataset and performance metrics used. It then presents the proposed concatenated deep learning algorithm, explaining the mathematical models of the CNN and GRU components. Finally, the performance evaluation section discusses the simulation results, comparing the proposed model against other techniques based on various performance indicators

2. Related Work

Berkeley researchers offer a design for a WSN that runs ceaselessly on renewable energy from the climate [7]. Two-stage storage systems made contained a rechargeable solar lithium battery and a supercapacitor are utilized in this method [8]. A gadget for harvesting human energy is built and examined, using a power hardware module to harness the energy created by a walking human [9]. Look into a physically autonomous sensor that uses mechanical nano energy derived from human beings to take advantage of movement location and physiological signal monitoring [10]. A wireless energy harvesting cognitive industrial internet of things (CIIOT) is proposed, which simultaneously performs range detecting and transmissions and harvests wireless energy from a primary client [11-13]. In any case, these research' energy-harvesting gadgets produce less energy and need movement, such mechanical vibrations in the system [14-15]. They cannot be broadly applied to industrial settings and have stayed in the laboratory design and verification stages.

Data gathering, transmission, and handling are a WSN's primary energy-consuming operations, and these activities are the vital targets of energy usage improvement [16]. Artificial neural networks are utilized to manage data sampling and preserve hub energy [17]. To increase the WSN's useful life, an adaptive sampling strategy that considers the temporal and spatial correlation of the sensor data is proposed. Created a ZigBee and Geiger Muller tube system that reasonably estimates radiation and temperature monitoring [18]. Improved uptime during power outages by settling on the ideal portable hub antenna configuration and the most efficient asset management strategy for relay hub selection. Suggested an approach to adaptive data collection that would reduce energy consumption and advance data transmission [19-20]. These methods, which are more commonly used in labs, can precisely control energy, reduce energy consumption, and lengthen the lifespan of systems. However, these methods have on occasion been employed in contexts including difficult outside conditions, such as present HVDC transmission line electromagnetic measurement systems.

China currently has a large number of HVDC transmission lines operational. The electromagnetic environment must be considered throughout the whole transmission line lifecycle, from design to installation and operation [21]. When studying the electromagnetic field, it is crucial to take into account the electric field beneath the (HVDC) transmission lines. With the help of a wireless sensor network (WSN) electric-field measurement system (EFMS), the electric field under the HVDC transmission cables can be observed. One important part of the EFMS and a popular area of study is the electric-field sensor. Optical, field factory, and MEMS sensors are just a few of the several possible techniques [22]. The charge enrolment guideline states that MEMS sensors convert the DC electric field into measurable electrical values using micromechanical resonators. One-way optical sensors detect electric fields is by using the electro-optic impact concept, which involves watching how a field changes the refractive index of a crystal [23]. The field factory sensors employ the widely recognized charging standard. At regular intervals, the engine opens and protects the electric field, keeping the rotor spinning at a constant speed. By cycling between charging and discharging charges, the enrolment cathode can provide an AC signal that is proportionate to the applied DC electric field. Power consumption of the EFMS is significantly increased due to the constant operation of the electric-field factory sensor's motor [24]. Since electric field sensors do not require an engine, they are able to utilize power-efficient MEMS and optical sensors. Unfortunately, most current electric-field measurement systems rely on electric-field factories as their electric-field sensors, mostly because of affordability and stability concerns [25].

In order to study the electric-field circulation under the transmission lines, a range of sensors can continuously measure the electric field. It is common practice to run EFMSs on batteries due to the lack of a reliable power source at many testing sites [26]. Because of the system's usual data advancement requirements and the battery capacity, extending the usable life of a battery-powered EFMS is necessary yet problematic. New features, like data transfer, are incorporated into the EFMS because of real-time checking and distributed sensor networks; nevertheless, this considerably raises the system's energy consumption. The major objective of this work is to provide an energy-efficient booking technique that may be used to extend the lifespan of an EFMS.

3. Research Methodology

In any situation where the Internet of Things is put to substantial use, no matter how large or little. Every single one of the domains makes use of the IoT's feature advantages, which greatly boost the performance of real-time applications. There is a pressing need for efficient data management due to the explosion of Internet of Things (IoT) devices, which has enabled smart cities and smart agriculture. It is possible that distributed computing can efficiently manage the IoT network in terms of storage and figure limitations. Internet of Things devices with low resources gather data and upload it to the cloud for processing. Due to the heterogeneity of the network, data transmissions from IoT devices to the cloud, whether incoming or departing, experience high latency and bandwidth needs. Edge computing is suggested as a means to lessen latency in IoT networks; it involves moving processing power from the cloud to the client end. The basic IoT edge analytics setup is illustrated in Figure 1, showing how raw sensor data is processed through edge gateways before being analysed in the cloud. As an add-on, edge registering is useful for cloud and Internet of Things networks. Distributed computing's figure load is drastically reduced in real-time applications by using edge registering.

In this study, the Intel Berkeley Research Laboratory WSN dataset was used for model evaluation. This dataset includes temperature and humidity sensor readings from 54 Mica2Dot devices deployed in an indoor environment. Prior to training, the data was cleaned, normalized, and segmented into input sequences suitable for time-series modelling. The proposed deep learning model was assessed using multiple performance metrics, including resource utilization, response time, execution time, average delay, and overall efficiency. These metrics were selected to directly reflect the real-time performance and energy efficiency of industrial WSN applications under edge-integrated IoT conditions.

With their three-layer heterogeneous design, cloud-edge cloud IoT networks require proper asset planning to further increase productivity and service quality. The diverse types of data collected by the Internet of Things network necessitate different approaches to data processing. All data should be handled on the edge or in the cloud when asset demands are booked. The asset requirements can be represented by tasks, and the edge network can obtain these demands in turn. After that, it selects the best cloud assets and arranges for IoT networks to use them for additional processing. To avoid SLA violations, cloud providers should think about things like load balancing, energy consumption, and bandwidth congestion, regardless of how many resources they offer.

In order to reduce the real-time real data handling latency in the dispersed computing environment, IoT networks use edge registration. The edge enhances the performance of the network and reduces compute, blockage, and data transmission delay by supplying the appropriate cloud assets to IoT networks. Distributed computing assets should be planned for edge figuring into Internet of Things networks as part of an effective asset planning strategy. Planning must take asset elasticity and scalability into account. When assets in the cloud are shared via edge networks, their scalability and flexibility change since most of these assets are virtual or physical. Because not all applications have the same asset demands, the edge should be aware of these needs when booking assets for IoT networks. This is because different applications require different registering assets.

The resource scheduling algorithms that have emerged recently are either machine learning- or statistically-based scheduling procedures. Based on the resource requirements, the best resources are chosen from the resource pool. Deep learning techniques have recently supplanted statistical and machine learning-based scheduling models in order to improve scheduling performance. When it comes to scheduling resources at the edge of a network, deep learning techniques like RL, Q-learning, and deep neural networks are heavily utilized. Cutting down on wait times and scheduling delays is critical for optimizing efficiency, even when deep learning methods work as expected. Asset planning in edge-integrated IoT networks is facilitated by this combined deep learning method.

Integrating IoT Services in the Cloud Many applications on the Internet of Things have embraced distributed computing because of its capacity to process, store, and analyse massive amounts of data. The distributed computing environment is another option for the Internet of Things (IoT) that relies on cloud computing to enable device connection. According to their requirements, customers can access cloud services whenever and wherever they like. This internet of things (IoT) approach that integrated the cloud was utilized by numerous smart city, transportation, agricultural, and healthcare applications. However, the data processing experiences latency due to the long-distance data transit between IoT devices to the cloud. Various applications of IoT are applied for actual real-time time operations, which should deliver rapid replies by evaluating the data. Delays in answers will harm the quality of services in IoT applications. Meanwhile, it is vital to maintain a steady connection between devices and the cloud49 which is also a challenge while integrating the cloud with IoT. Thus, it is vital to consider few insights as provided below when designing cloud- based integrated IoT apps.

- An exceptionally fast reaction time and as little delay as possible from beginning to end are essential for improving service quality.
- The cloud and IoT application hubs are dependent on a reliable and steady network.
- Significant processing complexity will result from adding more networking conventions. For this reason, limiting computational difficulties requires careful consideration while selecting conventions.

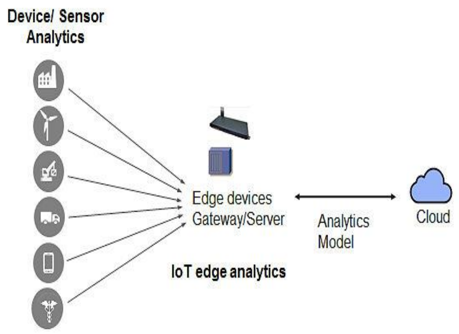

Figure 1. Overview of IoT Edge Analytics Architecture showing the flow of sensor data from devices to edge devices and the cloud.

A. Industrial Automation and the Need for WSNs

WSNs have played an important role in the development of industrial automation, especially within the context of Industry 4.0, which emphasizes the integration of cyber-physical systems, Internet of Things (IoT), and data-driven decision-making, so that in WSNs, through real-time monitoring and control of industrial processes, improvements in efficiency and productivity and safety can be facilitated [27]. One of the important applications includes environmental monitoring, condition monitoring of equipment, predictive maintenance, and process optimization where sensor nodes collect and transmit critical data to enhance operational intelligence [28]. Nevertheless, although WSNs can be deployed in industrial automation, challenges faced include energy consumption, network scalability, and data reliability in harsh environments. These challenges bring opportunities to innovation toward energy-efficient protocols, advanced sensor technologies, and robust communication strategies; they are crucial for optimizing the performance and sustainability of industrial systems in the industry 4.0 era. Additionally, Multi-Input Multi-Output (MIMO) communication techniques can enhance the scalability and reliability of WSNs in industrial automation, particularly in high-interference or dense deployment scenarios.

B. Energy Efficiency in Wireless Sensor Networks

Energy efficiency is very sensitive to WSNs, particularly for industrial applications in large-scale remote environments or in harsh environments where the nodes are set up [29]. The energy usage of WSNs directly influences the life of the battery-powered nodes, the reliability of networks, and the operational costs in general. In industrial environments, where the requirement is for high continuity monitoring of data, the challenge is how to balance these data transmission frequencies with the limited energy sources available on the sensor nodes [30]. This can be solved by using several strategies-including energy-efficient routing protocols, adaptive techniques of data transmission, and duty cycling. Techniques that involve data aggregation, energy harvesting, and the use of low-power communication protocols are also underway in reducing energy consumption but without sacrificing performance and reliability in industrial automation systems for WSNs [31].

C. Edge IoT Integration

An edge-integrated m module can help with the limitations of a cloud-integrated IoT ecosystem. Not only does edge registration reduce handling complexity and latency, but it also brings cloud resources directly to the user's device. Some examples of applications of edge figuring include cloudlets, flexible edge processing, and mist registering. Each of these methods reduces the amount of time data must be processed before IoT applications get their responses. Reduced latency and data transfer times are the results of edge processing, which places assets in close proximity to the Internet of Things. The IoT app can take advantage of edge processing to improve both its local and transmitted data handling capabilities. Edge enhances system resilience and fault tolerance by reducing bandwidth requirements and providing Internet of Things clients with great adaptability. A layered view of the cloud-edge-device architecture is shown in Figure 2, demonstrating how edge nodes handle local processing and interact with the cloud for more complex tasks.

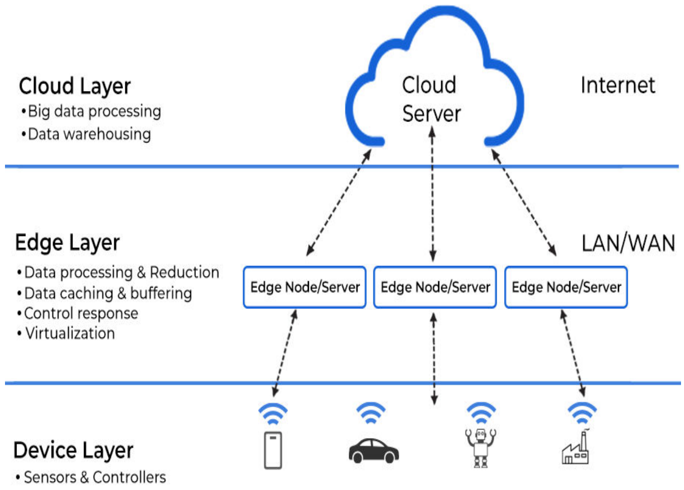

Figure 2. Layered architecture of Cloud-Edge-IoT integration illustrating data processing, communication flow, and control from device to cloud via edge nodes.

4. Proposed Concatenated Deep Learning Algorithm

An in-depth mathematical model of the concatenated deep learning technique that has been suggested is presented in this section. A convolutional neural network and a gated recurrent unit are utilized for initial feature extraction in the most basic form of the proposed model, as illustrated in Figure 3. The asset's category, sub-class, class, duration, and other relevant criteria are taken into account when determining the demand for the asset.

Asset needs can be parsed into local and regional details by use of a one-dimensional convolutional neural network. Similar to how features are recovered in the latter stage of the process using a gated recurrent unit, the best asset for planning in edge processing is chosen in the first stage. For this project, we've settled on a gated recurrent unit (GRU) because of its ease of use and high performance. When compared to conventional long short-term memory (LSTM), GRU performs better due to its faster input processing and less parameter usage. By fixing the vanishing gradient issue in RNN, GRU could make current booking algorithms better. Finally, after the combined traits are categorized, the best resources for the jobs are scheduled.

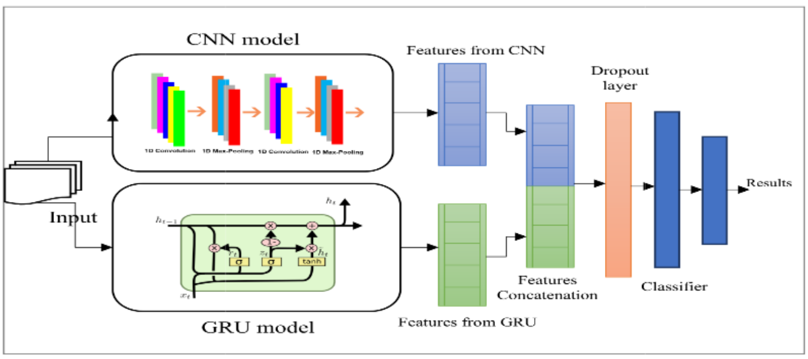

Figure 3. An extended deep learning model that has been proposed

- **Gated Recurrent Unit**

As a GRU model, a gated recurrent neural network is recommended. With only two gates, GRU stands in stark contrast to LSTM's trio. The GRU’s update and reset gates not only improve union rates but also reduce the number of parameters needed compared to an LSTM. Using its memory cell, the GRU model may retrieve crucial data and identify situations in the input asset requirements. The GRU's reset gate forgets or erases the redundant data. The GRU model often takes period series data as input, even if the asset demand input is typically period grouping data with a single time step. The activation is successful, and the GRU model's outputs are obtained. By feeding the principal layer's output into the next layer and repeating the process, we may extract the important features from the input to the resultant layer. Mathematical descriptions of the GRU model are as:

$$\mathcal{G}_u = \sigma\left(w_u\left(\tilde{v}^{(t-1)}, x^{(t)}\right) + b_u\right) \quad \textbf{(1)}$$

$$\mathcal{G}_r = \sigma\left(w_r\left(\tilde{v}^{(t-1)}, x^{(t)}\right) + b_r\right) \quad \textbf{(2)}$$

Here, Gu handles the update gate while Gr handles the reset gate. Update gates differ from reset gates in that their range is [0,1] rather than [-1,1]. Wr denotes the capability to reset the gate weight and Wu stands for the capability to update the gate weight. Just as how Br takes care of the update gates bias vector, as does as for the reset gate. The candidate activation capability for the recurrent unit is created using the following formula, which is based on the gate works.

$$\tilde{v}^{(t)} = tanh\left[w_u\left(\mathcal{G}_r \times \tilde{v}^{(t-1)}, x^{(t)}\right) + b_u\right] \quad \textbf{(3)}$$

To handle the bias vector, the activation capacity weight factors—Wu for the update gate—are applied when the input training data is labeled as X(t). The final step is to transmit the GRU model's output as:

$$v^{(t)} = \left((1 - \mathcal{G}_u) \times \tilde{v}^{(t-1)}\right) + \left(\mathcal{G}_u \times \tilde{v}^{(t)}\right) \quad \textbf{(4)}$$

where d is a function of the output of the prior unit and v(t-1) is the input of the present unit. Merging the output characteristics of the CNN and GRU models results in extra processing steps that determine which assets are most suitable for booking.

- **Convolutional Neural Network**

The input is sorted into subclasses according to the specifications by the Convolutional Neural Network Model employed in the proposed work before the Convolution layer takes over. Two max-pooling layers and two convolution layers are employed by the suggested design to glean useful details from the asset needs. In terms of data handling and maintaining local interactions, CNN outperforms traditional neural network models. Compared to earlier models, this neural network method conveys attributes more clearly while jellying the input data's spatial localization. Through autonomous training, the network is able to absorb various data qualities. The CNN module is based on the correlated cycle principle of the convolution interaction. To construct the convolution interaction correctly, one needs to think about the loads of the one-dimensional aspects component, which are represented by the words {W1,W2,...,Wn}, where n is the kernel length.

$$y_t = f\left(\sum_{i=1}^{n} w_i * x_{t-i+1}\right) \quad \textbf{(5)}$$

We say that the data created at time t is Yt and that the input sample is Xt. The proposed model makes use of the Redressed Linear Unit (RELU) as its activation capability. The activation capability can be mathematically expressed as

$$ReLU(x) = \begin{cases} x, & x > 0 \\ 0, & x \le 0 \end{cases} \quad \textbf{(6)}$$

The suggested architecture employs max pooling to cap the feature size subsequent to the convolution layer. The outputs of the convolution layer are down-sampled to reduce unpredictability. Maximum pooling operator mathematically expresses the forwarding of the maximum value as

$$\mathcal{P}_{j.m} = \max h_{j,(m-1)}\, n + r \quad \textbf{(7)}$$

where n is the permitted area-to-area pooling shift, m is the maximum pooled band, and j are the channels. With most convolution bands, the pooling layer reduces their dimensionality. Batch normalization follows pooling capabilities and improves training results by standardizing the features. The mathematical description of batch normalization characteristics is up next.

$$u = \frac{1}{n_{bat}} \sum_{n=1}^{n_{bat}} x_n$$

$$\sigma^2 = \frac{1}{n_{bat}} \sum_{n=1}^{n_{bat}} (x_n - \mu)^2$$

$$\hat{x} = \frac{x_n - \mu}{\sqrt{\sigma^2 + \varepsilon}}$$

$$\boldsymbol{y_n = \gamma \hat{x}_n + \beta} \quad \textbf{(8)}$$

where X n is the input data and N bat is the batch size. While σ2 addresses the batch variance, μ indicates the mean. In order to avoid zero gradients, the normalized data is associated with a constant ε, denoted as X̂. Dand K is the graphic depiction of the learning vector parameters. The features that are represented by the output are ʃ and β. Yn stands for the feature that is produced.

Afterwards, the CNN and GRU models' properties are integrated. To avoid overfitting the data, apply a dropout layer following concatenation. Finally, the collected features are classified using the fully linked network layer and SoftMax algorithms in order to assign the correct resources to a job. The SoftMax capability can be stated numerically as

$$\boldsymbol{\hat{y} = softmax(Q)} \quad \textbf{(9)}$$

the result of the dropout layer is denoted by Q. Last but not least, the mistake capacity of the suggested model is checked using a cross-entropy capability. Mathematically, it's expressed as

$$\boldsymbol{\ell = -\frac{1}{b}\sum_{i=1}^{n} y_i \log y_i'} \quad \textbf{(10)}$$

As an example, yi' represents the expected component and yi addresses the actual component, whereas b, n, and yi are the sizes of the batch and training samples, respectively.

5. Performance Evaluation

We empirically validate the performance of the recommended deep learning model by incorporating the package and works as capabilities in a Python simulation study. Using these tools, hyperparameters are automatically generated and adjusted to enhance performance even more. The benchmark data used in the investigation comes from Intel's Berkeley research laboratories and contains 96. For the purpose of this simulation investigation, the hyperparameters used are detailed in Table 1. Validation is conducted by examining and comparing current strategies, including the hereditary algorithm, the Improved Particle Swarm Optimization (IPSO) algorithm, Long Short-Term Memory (LSTM), and the Bidirectional Recurrent Neural Network (BRNN).

Table 1: Hyperparameters of the suggested DL technique

S. No	Parameters	Value
1	Conv filters 1	32
2	Conv filters 2	128
3	GRU Units	64
4	Drop out	0.0
5	Epochs	25
6	Batch Size	64

The proposed model's accuracy and misfortune curves are shown in Figure 4. Measuring performance is based on the standard method of testing and training. Each dataset is partitioned for testing, validation, and training at 70:20:10. The results have not altered after more than 25 generations of measurement. According to the results, the suggested model is the most accurate, and it has been verified.

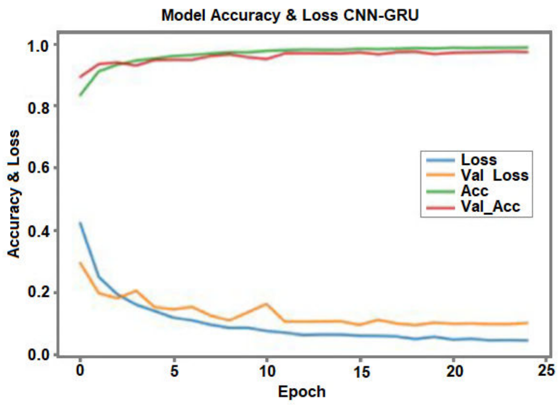

Figure 4. Accuracy & Loss

The suggested model's asset use is compared to the ongoing models in Figure 5. The results show that the suggested model is able to make the most efficient use of assets thanks to the optimal choice of assets. Positions are allocated optimal assets, which expedites data processing and frees up these assets for other uses. The overall asset consumption of the suggested approach is thus larger than that of current strategies. The suggested model and the existing BRNN models perform similarly, however other models display large discrepancies in the values of asset utilization.

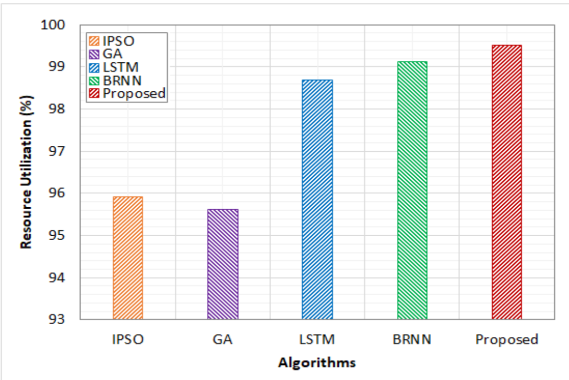

Figure 5. Resource utilization (%)

The response times of the ongoing asset planning methods and the suggested concatenated deep learning strategy are contrasted in Figure 6. Reaction time is the amount of time it takes for the planning algorithm to assess and plan for asset demands. The average time is calculated for each strategy that requests a certain asset from edge figuring. The suggested paradigm for asset demands demonstrates a minimal reaction time of 1.25 seconds. However, when other methods are employed, the average rises. By completing asset demands in 1.66s, 1.98s, and 2.20s, respectively, the LSTM model, the BRNN model, GA, and IPSO all outperform the suggested model in terms of reaction time.

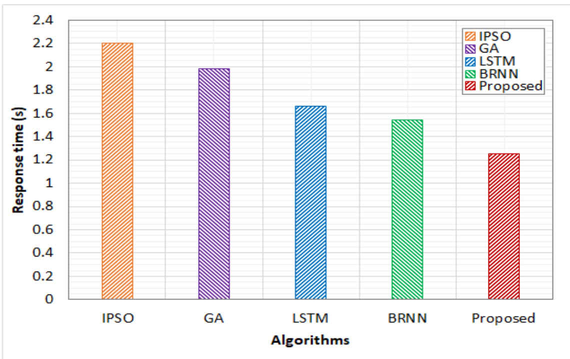

Figure 6. Response time (s)

The suggested model's total execution seasons are compared to those of the state-of-the-art models in Figure 7. Execution time includes the time required to evaluate asset demand, choose the best asset from the pool, and schedule that asset. According to the results, the suggested model has the fastest execution time when compared to alternative strategies for planning. Compared to the BRNN model, the LSTM-based booking, the GA, and the IPSO model, the recommended model's execution season of 10.25s is 5s quicker, 8s faster, 11s faster, and 16s faster, respectively. We prioritize any timesaving's that may be achieved in the asset booking procedure for edge registration. The suggested model's performance is validated by considering the average delay given by both the present and new models. This is done through research into different asset demands.

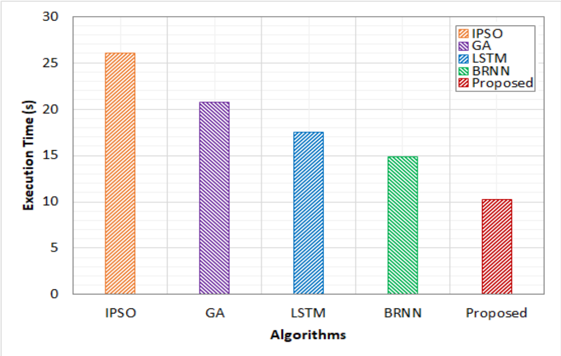

Figure 7. Temporal Analysis of Execution

The results, shown in Figure 8, show that compared to the state-of-the-art methods, the suggested model minimizes delays more effectively on average. By a margin of five seconds, the IPSO model outperforms the suggested model in terms of latency. When compared to GA-based booking, the suggested process is 4.3 seconds slower. Models trained with LSTM and BRNN outperform those using GA and IPSO by a small margin. However, this model is far from perfect. The LSTM-based booking model is 2 seconds off and the BRNN model is 1.5 seconds off; the recommended model has a minimum delay of 1.15 seconds.

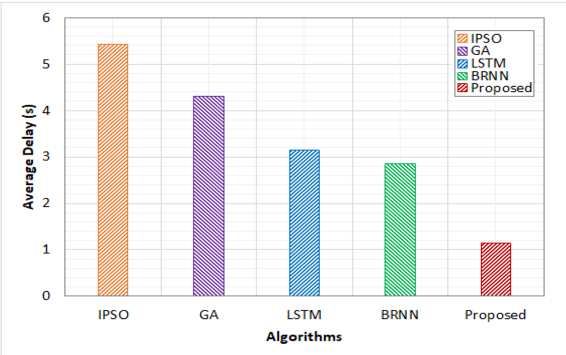

Figure 8. Analysis of Average Delays

Figure 9 displays an effective comparison of booking algorithms. By comparing the proposed and current models' execution times, reaction times, and delay factors, we may get a sense of how efficient they are overall. The proposed model outperforms the competition across the board, improving the efficiency of both the IoT networks and the edge-processing platform. Compared to the current booking methods, the suggested model has a substantially higher maximum efficacy of 99.48%.

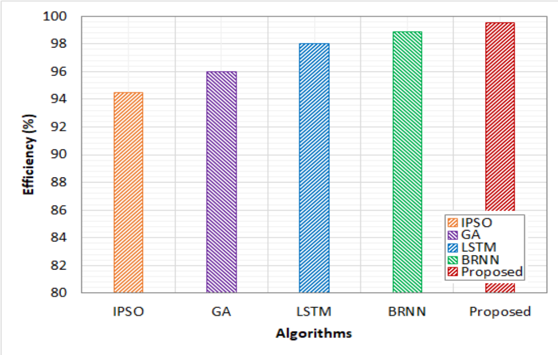

Figure 9. Efficiency Analysis

Table2 provides an overview of the overall performance metrics used to compare the proposed model to the current models. The results demonstrate that the proposed approach achieves a higher level of asset utilization and productivity when compared to other existing booking systems. The proposed model also has the quickest execution and reaction times, making it ideal for real-time applications that need to record data produced or received efficiently through asset allocation.

Table 2: Analysing Performance in Comparison

Methods	Resource Utilization	Response Time	Execution Time	Average Delay	Efficiency
IPSO	95.8217%	2.2123s	26.0035s	5.4136s	94.4532%
GA	95.7245%	1.9742s	20.6379s	4.2948s	96.1037%
LSTM	98.5871%	1.6734s	17.4659s	3.1368s	97.9082%
BRNN	99.0213%	1.5529s	14.8164s	2.8547s	98.8045%
Proposed	99.5234%	1.2531s	10.2537s	1.1539s	99.4862%

Efficiency of the cycle is supported by the suggested model's best asset utilization, base execution, and reaction times when compared to current strategies.

6. Discussion

The proposed deep learning-based resource-scheduling model demonstrates high efficiency in energy-sensitive industrial environments. Its architecture, which combines CNN and GRU, allows it to learn both spatial features and temporal demand patterns, leading to improvements in response time and execution speed. This efficiency is particularly beneficial for large-scale WSN deployments, where traditional heuristics or centralized schedulers become bottlenecks. The model's performance remains consistent even as the number of sensor nodes increases, due to its ability to dynamically adapt asset allocation based on learned patterns.

One critical aspect of WSN design in industrial automation is managing the **trade-off between energy efficiency and latency**. While energy-saving mechanisms like duty cycling can introduce delays, the proposed model minimizes this through predictive scheduling — thus achieving a balance that supports real-time applications without compromising battery life.

Furthermore, **fault tolerance** in harsh industrial environments is a significant concern. Although not explicitly modeled, the system's decentralized asset planning and real-time adaptation allow for continued functionality even when individual nodes fail or drop out, improving resilience. Future extensions could incorporate self-healing mechanisms or use ensemble models to further boost robustness.

Lastly, integration **with energy harvesting technologies** (such as vibration-based or solar charging systems) could extend the operational life of the network. The scheduling model can be adapted to prioritize tasks based on residual energy or harvested power availability, contributing to sustainable industrial IoT systems.

7. Conclusion

Here, we provide a mixed-methods deep learning approach to asset planning in IoT networks with embedded edges. In order to prioritize characteristics based on asset demands during the asset-planning phase, the suggested work utilizes a gated recurrent unit and a one-dimensional convolutional neural network. The optimal planning assets are identified through the application of deep learning models that quickly assess and combine time-series requests, followed by classification. With the use of simulation analysis, we can see how the suggested model stacks up against other methods in terms of effectiveness, average delay, reaction time, execution time, and asset utilization, as well as against Genetic Algorithm (GA), LSTM, BRNN, and Genetic Algorithm (GA). The overall

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] Chehri, R. Saadane, N. Hakem, and H. Chaibi, "Enhancing energy efficiency of wireless sensor network for mining industry applications," *Procedia Comput. Sci.*, vol. 176, pp. 261–270, 2020.

[2] S. Ahmed, M. Z. Chowdhury, and Y. M. Jang, "Energy-efficient UAV-to-user scheduling to maximize throughput in wireless networks," *IEEE Access*, vol. 8, pp. 21215–21225, 2020.

[3] O. Alamu, A. Gbenga-Ilori, M. Adelabu, A. Imoize, and O. Ladipo, "Energy efficiency techniques in ultra-dense wireless heterogeneous networks: An overview and outlook," *Eng. Sci. Technol. Int. J.*, vol. 23, no. 6, pp. 1308–1326, 2020.

[4] T. A. Alghamdi, "A hybrid algorithm for energy-efficient cluster head selection in wireless sensor networks," *IEEE Access*, vol. 8, pp. 189211–189222, 2020.

[5] V. K. Arora, V. Sharma, and M. Sachdeva, "A survey on LEACH and other routing protocols in wireless sensor networks," *Optik*, vol. 127, no. 16, pp. 6590–6600, 2016.

[6] Bagwari, J. Logeshwaran, K. Usha, R. Kannadasan, M. H. Alsharif, P. Uthansakul, and M. Uthansakul, "An Enhanced Energy Optimization Model for Industrial Wireless Sensor Networks Using Machine Learning," *IEEE Access*, vol. 11, pp. 103377–103391, 2023.

[7] J. Bhola, S. Soni, and G. K. Cheema, "Genetic algorithm based optimized LEACH protocol for energy efficient wireless sensor networks," *J. Ambient Intell. Humaniz. Comput.*, vol. 11, pp. 1281–1288, 2020.

[8] S. Abella *et al.*, "Autonomous energy-efficient wireless sensor network platform for home/office automation," *IEEE Sens. J.*, vol. 19, no. 9, pp. 3501–3512, 2019.

[9] S. M. Chowdhury and A. Hossain, "Different energy saving schemes in wireless sensor networks: A survey," *Wirel. Pers. Commun.*, vol. 114, no. 3, pp. 2043–2062, 2020.

[10] H. Kathiriya, A. Pandya, V. Dubay, and A. Bavarva, "State of art: energy efficient protocols for self-powered wireless sensor network in IIoT to support industry 4.0," in *Proc. 8th Int. Conf. Rel., Infocom Technol. Optim. (Trends Future Directions) (ICRITO)*, 2020, pp. 1311–1314.

[11] J. Blanco, A. García, and J. D. L. Morenas, "Design and implementation of a wireless sensor and actuator network to support the intelligent control of efficient energy usage," *Sensors*, vol. 18, no. 6, p. 1892, 2018.

[12] J. Li, J. Lv, P. Zhao, Y. Sun, H. Yuan, and H. Xu, "Research and application of energy-efficient management approach for wireless sensor networks," *Sensors*, vol. 23, no. 3, p. 1567, 2023.

[13] R. Jin, X. He, and H. Dai, "Communication efficient federated learning with energy awareness over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 5204–5219, 2022.

[14] K. Karunanithy and B. Velusamy, "Cluster-tree based energy efficient data gathering protocol for industrial automation using WSNs and IoT," *J. Ind. Inf. Integr.*, vol. 19, p. 100156, 2020.

[15] U. K. Lilhore, O. I. Khalaf, S. Simaiya, C. A. Tavera Romero, G. M. Abdulsahib, and D. Kumar, "A depth controlled and energy-efficient routing protocol for underwater wireless sensor networks," *Int. J. Distrib. Sensor Netw*, vol. 18, no. 9, 2022.

[16] S. N. Mohanty and M. R. Patra, "Deep Learning-Based Distributed Data Mining (DDM) model for energy-efficient wireless sensor networks," *IEEE Access*, vol. 8, pp. 67890–67901, 2020.

[17] S. N. Mohanty, E. L. Lydia, M. Elhoseny, M. M. G. Al Otaibi, and K. Shankar, "Deep learning with LSTM based distributed data mining model for energy efficient wireless sensor networks," *Phys. Commun.*, vol. 40, p. 101097, 2020.

[18] Nakas, D. Kandris, and G. Visvardis, "Energy efficient routing in wireless sensor networks: A comprehensive survey," *Algorithms*, vol. 13, no. 3, p. 72, 2020.

[19] J. O. Ogbebor, A. L. Imoize, and A. A. A. Atayero, "Energy efficient design techniques in next-generation wireless communication networks: emerging trends and future directions," *Wirel. Commun. Mobile Comput.*, vol. 2020, p. 1–19, 2020.

[20] P. Sreedevi and S. Venkateswarlu, "Comparative analysis of energy efficient routing protocols with optimization in WSN," *Int. J. Interact. Des. Manuf.*, pp. 1–16, 2022.

[21] R. Tanash, M. AlQudah, and S. Al-Agtash, "Enhancing energy efficiency of IEEE 802.15.4-based industrial wireless sensor networks," *J. Ind. Inf. Integr.*, vol. 33, p. 100460, 2023.

[22] S. Sivakumar, J. Logeshwaran, R. Kannadasan, M. Faheem, and D. Ravikumar, "A novel energy optimization framework to enhance the performance of sensor nodes in Industry 4.0," *Energy Sci. Eng.*, vol. 12, no. 3, pp. 835–859, 2024.

[23] M. Sahoo, T. Amgoth, and H. M. Pandey, "Particle swarm optimization based energy efficient clustering and sink mobility in heterogeneous wireless sensor networks," *Ad Hoc Netw.*, vol. 106, p. 102237, 2020.

[24] T. M. Behera *et al.*, "Energy-efficient routing protocols for wireless sensor networks: Architectures, strategies, and performance," *Electronics*, vol. 11, no. 15, p. 2282, 2022.

[25] M. Tinatin, Q. Guibin, and Z. Jinghui, "Evolution of wireless communication network and its impact on human life," *J. Comput. Sci. Technol.*, vol. 27, no. 5, pp. 857–868, 2012.

[26] W. Mao, Z. Zhao, Z. Chang, G. Min, and W. Gao, "Energy-efficient industrial Internet of Things: Overview and open issues," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7225–7237, 2021.

[27] T. Wang, X. Li, Y. Liu, and Y. Zhang, "Fog-based data collection for wireless sensor networks with mobile sinks," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 850–859, 2017.

[28] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, 2020.

[29] Z. Yang, M. Chen, W. Saad, W. Xu, M. Shikh-Bahaei, H. V. Poor, and S. Cui, "Energy-efficient wireless communications with distributed reconfigurable intelligent surfaces," *IEEE Trans. Wireless Commun.*, vol. 21, no. 1, pp. 665–679, 2021.

[30] Yousaf, F. Ahmad, S. Hamid, and F. Khan, "Performance comparison of various LEACH protocols in Wireless Sensor Networks," in *Proc. IEEE 15th Int. Colloq. Signal Process. It is Appl. (CSPA)*, 2019, pp. 108–113.

[31] X. Zhuo, M. Liu, Y. Wei, G. Yu, F. Qu, and R. Sun, "AUV-aided energy-efficient data collection in underwater acoustic sensor networks," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10010–10022, 2020.