Digital Twin-Driven Scheduling in IEEE 802.15.4e TSCH Networks

Md. Niaz Morshedul Haque
Assistant Professor
Department of Electrical and Electronic Engineering
Leading University, Sylhet

ABSTRACT
IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) is widely used in Industrial Internet of Things (IIoT) systems due to its reliability and energy-efficient communication. However, efficient scheduling in TSCH networks remains a challenging task, particularly in balancing throughput and delay. In this paper, a digital twin (DT)-based scheduling framework is proposed to address this issue. The scheduling problem is formulated as a maximum-weight bipartite matching model, where link-to-cell assignments are optimized using the Hungarian algorithm within the digital twin environment. The DT replicates the behavior of the physical network and generates labeled data, which is used to train a deep neural network (DNN) for fast scheduling decisions. Simulation results demonstrate that the proposed approach achieves near-optimal performance compared to the Hungarian method while significantly reducing computational complexity. The results highlight the effectiveness of digital twin technology for efficient and scalable TSCH network management.

General Terms
Algorithms, Network Scheduling, Wireless Communication, Industrial IoT

Keywords
Digital Twin, TSCH, IEEE 802.15.4e, Network Scheduling, Deep Learning, Industrial IoT

1. INTRODUCTION
The Internet of Things (IoT) has gained significant attention due to its ability to enable intelligent connectivity among devices for various applications [1]. The Industrial Internet of Things (IIoT), as an extension of IoT, facilitates real-time monitoring, automation, and control in industrial environments [2]. To support reliable and energy-efficient communication in such systems, the IEEE 802.15.4e standard introduces Time-Slotted Channel Hopping (TSCH), which combines time-slotted access with channel hopping to ensure high reliability and low power consumption [3], [4].In TSCH networks, scheduling plays a crucial role in determining how communication links are assigned to time slots and channels. Efficient scheduling directly impacts network performance in terms of throughput, delay, and reliability [6]–[9]. However, designing optimal scheduling algorithms remains challenging, particularly under dynamic wireless conditions where channel state information (CSI) is uncertain or time-varying [7], [9]. Traditional scheduling approaches often rely on optimization techniques that assume either instantaneous or statistical channel information. While these methods can achieve optimal or near-optimal performance, they typically involve high computational complexity and are not suitable for real-time applications [7], [8]. This limitation motivates the need for efficient and scalable alternatives.

Recently, digital twin (DT) technology has emerged as a promising solution for modeling and optimizing complex systems. A digital twin is a virtual representation of a physical system that can replicate its behavior and support data-driven decision-making [10]–[12]. In wireless networks, DT enables the simulation of network dynamics and provides a platform for generating large-scale training data without requiring real-world deployment [13]–[16].In parallel, deep learning techniques have demonstrated strong capability in approximating complex optimization problems in wireless communications [17]–[20]. By learning the mapping between system inputs and optimal outputs, deep neural networks can provide fast and efficient solutions with reduced computational cost.

Motivated by these advancements, this paper proposes a digital twin-based scheduling framework for IEEE 802.15.4e TSCH networks. The scheduling problem is formulated as a maximum-weight bipartite matching problem, where link-to-cell assignments are optimized based on throughput and delay. The Hungarian algorithm is employed within the digital twin to generate optimal scheduling decisions, which are then used to train a deep neural network for efficient approximation [21], [22].

The main contributions of this paper are summarized as follows:

- A digital twin framework is developed to model TSCH network behavior and generate labeled scheduling data.
- A learning-based scheduling approach is proposed to approximate optimal solutions using a deep neural network.
- The proposed method achieves near-optimal performance with significantly reduced computational complexity compared to conventional approaches.

The remainder of the paper is organized as follows. Section 2 system model. Section 3 presents digital twin-based frameworks. Section 4 describes the proposed digital twin-based learning scheduling scheme. Section 5 discusses the performance evaluation and Section 6 concludes the paper.

2. SYSTEM MODEL
In this section, the system model of the TSCH network is presented, including scheduling, network structure, traffic assumptions, and channel characteristics. These components are used to formulate the scheduling problem.

2.1 TSCH Scheduling
In IEEE 802.15.4e TSCH networks, scheduling defines how communication links are assigned to time slots and channel offsets. Each communication opportunity, referred to as a cell, is identified by a pair of time slot and channel offset. Efficient

scheduling ensures reliable data transmission while minimizing interference.

Channel hopping is achieved by combining the absolute slot number (ASN) with the channel offset, allowing nodes to switch frequencies in a pseudo-random manner. The operating frequency can be expressed as:

$$f = \mathcal{F}[(ASN + Ch_{offset}) mod\ n_{ch}] \quad (1)$$

Where Ch_{offset} represents the channel offset, n_{ch} is the number of available channels and $\mathcal{F}$ denotes the mapping function.

2.2 Network Model

A centralized TSCH network consisting of one gateway and multiple sensor nodes is considered. The gateway is responsible for network synchronization and scheduling decisions. The network is represented as a graph $G = (V, M)$ where V denotes the set of nodes and M represents the set of communication links.

Each node is equipped with a half-duplex radio, meaning it can either transmit or receive at a given time but not both simultaneously. The scheduling algorithm assigns time slots and channels to links such that interference is avoided and communication efficiency is maintained.

A slot-frame is composed of multiple time slots that repeat periodically. Within each slot, links that do not interfere with each other can be scheduled simultaneously.

Fig. 2 illustrates a representative TSCH network topology along with its corresponding slot-channel scheduling matrix. The left side shows the network graph, where nodes are interconnected through communication links. The right side presents the scheduling structure, where time slots and channel offsets define communication cells. Non-interfering links are assigned to the same cell, enabling efficient utilization of network resources while avoiding collisions.

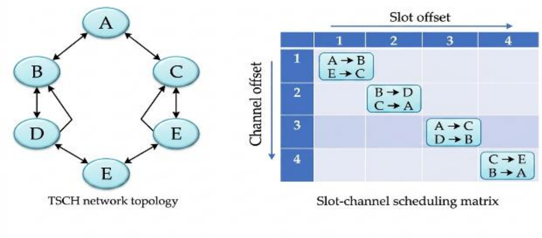

Fig 1: Example of TSCH network topology and corresponding slot-channel scheduling matrix.

2.3 Traffic Model

A saturated traffic model is assumed, where each node continuously has data available for transmission. This assumption simplifies the analysis and allows the scheduling algorithm to focus on maximizing throughput and minimizing delay.

Since wireless channel conditions vary over time, scheduling decisions are based on average link performance rather than instantaneous channel state information. The link weight is defined using both throughput and delay metrics.

2.4 Channel Model

Wireless channels are subject to fading and noise. In this study, Rayleigh fading is considered to model channel variations. The channel gain for a given link and cell is expressed as:

$$x_{m,c}(n) = |H_{m,c}(n)|^2 \quad (2)$$

where $H_{m,c}(n)$ represents the channel gain.

The achievable throughput depends on the signal-to-noise ratio and can be estimated using standard communication models. Delay is determined based on the position of the assigned cell within the slot-frame. These two parameters are combined to define the link weight used in scheduling.

2.5 Collision Model

To prevent interference, a collision graph $Q = (M, E)$ is defined, where each vertex represents a communication link and edges indicate potential conflicts between links.

Two links are considered interfering if they share a common node or operate within interfering transmission ranges. A valid schedule ensures that interfering links are not assigned to the same cell. Therefore, only non-conflicting links can be scheduled simultaneously.

Fig. 2 shows the collision graph of the TSCH network, where each node represents a communication link and edges indicate interference between links. Two links connected by an edge cannot be scheduled in the same cell. Therefore, scheduling must ensure that only non-interfering links are assigned to the same time slot and channel offset.

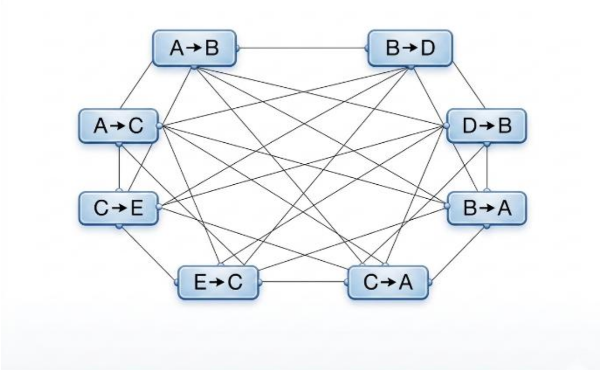

Fig 2: Collision graph representing interference relationships among TSCH links.

3. DIGITAL TWIN-BASED FRAMEWORK

A digital twin (DT)-based framework is proposed to enable efficient scheduling in IEEE 802.15.4e TSCH networks. The digital twin acts as a virtual representation of the physical network and allows data-driven optimization.

The real TSCH network provides information such as topology, link conditions, throughput, and delay. Using this information, the DT constructs a bipartite graph model, where communication links and available cells form two disjoint sets. The DT continuously updates its internal state based on network variations, enabling realistic data generation for scheduling decisions.

Fig. 3 illustrates the proposed digital twin-based scheduling framework. The real TSCH network provides link information such as throughput and delay, which is used to construct a bipartite graph. The digital twin replicates network behavior and generates optimal scheduling data using the Hungarian algorithm. A deep neural network (DNN) is trained based on this data to efficiently predict link-to-cell assignments. The model is continuously updated according to variations in network conditions.

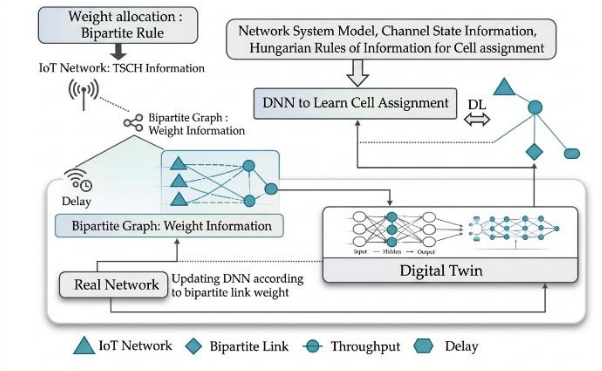

Fig 3: Digital twin-based scheduling framework for TSCH networks.

3.1 Bipartite Graph Model

The scheduling problem in TSCH networks can be modeled as a weighted bipartite graph, where the objective is to maximize the overall network performance. The graph is defined as $B = (M, C, E)$ where, M represents the set of communication links and C denotes the set of available cells in the slot-frame. Each edge $(\mathbf{m}, \mathbf{c}) \in \mathbf{E}$ connects a link to a cell. A valid assignment ensures that each link is allocated to only one cell, while avoiding conflicts among links. The weight of each edge is determined based on network performance metrics, specifically throughput and delay.

The edge weight for a link–cell pair is defined as a weighted combination of normalized throughput and delay:

$$W_{m,c}(n) = \alpha u_m^{N0} \theta_{m,c}^N(n) + (1 - \alpha) u_m^{N1} \psi_{m,c,n}^N \qquad (3)$$

where, α is a weighting factor that balances the importance of throughput and delay.

To ensure fairness, moving average values of throughput and delay are considered. These values are normalized to maintain stability across different network conditions.

The scheduling objective is to determine an optimal assignment that maximizes the total weight of the bipartite graph, which can be expressed as:

$$W_T * = \max \sum (m \in M) \sum (c \in C) \xi_{m,c} \; W_{m,c} \qquad (4)$$

where $\xi_{m,c}$ is a binary decision variable indicating whether a link is assigned to a specific cell

3.2 Hungarian Algorithm and Solution Technique

To obtain the optimal link-to-cell assignment, the Hungarian algorithm is applied to the weighted bipartite graph. The Hungarian method is a well-known combinatorial optimization technique that solves assignment problems in polynomial time.

The edge weights of the bipartite graph are defined based on throughput and delay metrics. Using these weights, the Hungarian algorithm determines the optimal assignment that maximizes the total network weight while ensuring that each link is assigned to only one cell.

The algorithm operates by transforming the weight matrix and iteratively refining row and column values to identify the optimal matching. This process guarantees a conflict-free assignment with maximum efficiency.

The resulting optimal assignments are used as reference solutions within the digital twin framework and serve as labeled data for training the learning-based scheduling model.

3.3 Dataset Generation Using Hungarian Algorithm

To train the learning-based model, a dataset is generated using the digital twin environment. The Hungarian algorithm is employed to obtain optimal scheduling decisions, which serve as labeled data for neural network training.

The dataset generation process considers channel variations, throughput, and delay to construct weighted bipartite graphs. These weights are computed using the formulation in (3), and the optimal assignments are obtained by maximizing the total weight.

Algorithm 1 Dataset Generation for Neural Network Training

Initialize: Moving average throughputu_m^{N0} and delay u_m^{N1}
//initial loop
Generate channel gain using Rayleigh distribution
Compute channel state $x_{m,c}(n)$
Calculate throughput $\theta_{m,c}^N(n)$ and delay $\psi_{m,c,n}^N$
Normalize throughput and update moving average values
Compute edge weight $W_{m,c}(n)$ using (3)
Apply Hungarian algorithm to obtain optimal assignment $W_T *$
//main loop
In this process, generate enough data samples for DNN training

4. DIGITAL TWIN-BASED LEARNING SCHEDULING SCHEME

In this section, a supervised deep neural network (DNN) is proposed to learn the optimal link-to-cell assignment in TSCH networks. The objective is to approximate the optimal scheduling solution obtained from the Hungarian algorithm under varying channel conditions.

The DNN is trained using the dataset generated by the digital twin framework, which reflects realistic network behavior. By learning the mapping between input features and optimal assignments, the proposed model enables fast and efficient scheduling with reduced computational complexity.

4.1 Deep Neural Network (DNN) Structure

A multilayer perceptron (MLP) is adopted as the learning model for scheduling. The network consists of an input layer, multiple hidden layers, and an output layer.

The input to the DNN is the set of bipartite edge weights $W_{m,c}(n)$, representing the relationship between links and cells for each data frame. The output of the network is the optimal cell assignment for each link, obtained from the Hungarian algorithm.

The number of input nodes corresponds to the total number of link–cell pairs, while the output nodes represent the assigned cells for all links. The hidden layers are designed to capture the nonlinear relationship between input features and optimal scheduling decisions.

The output of each hidden layer is computed using a nonlinear activation function. In this work, the Rectified Linear Unit (ReLU) activation function is employed due to its simplicity and computational efficiency. The activation function is defined as:

$$g(x) = max(0, x) \quad (5)$$

Fig. 4 illustrates the structure of the proposed deep neural network model. The network consists of an input layer, multiple hidden layers, and an output layer. The input layer represents the bipartite edge weights, while the output layer provides the corresponding cell assignments. The hidden layers learn the nonlinear relationship between input features and optimal scheduling decisions.

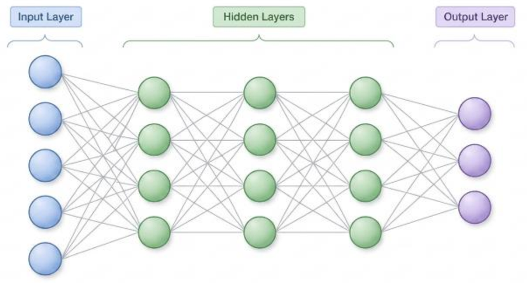

Fig 4: Structure of the proposed deep neural network for scheduling..

4.2 Neural Network Training and Digital Twin Integration

Training is a critical component of the proposed deep learning-based scheduling scheme. The objective is to enable the neural network to learn the mapping between bipartite edge weights and optimal link-to-cell assignments.

A supervised learning approach is adopted, where the training dataset is generated using the Hungarian algorithm within the digital twin environment. The dataset consists of input features (edge weights) and corresponding optimal assignments. The neural network is trained to approximate this mapping efficiently.

The training process aims to minimize the difference between the predicted assignment and the optimal solution. For this purpose, the mean squared error (MSE) loss function is employed:

$$J(W) = E(|c^* - \widehat{c^*}|)^2 \quad (6)$$

where c^* represents the optimal assignment and $\widehat{c^*}$ denotes the predicted output of the DNN.

The Adam optimizer is used to minimize the loss function due to its efficiency and fast convergence. A total of 10,000 data samples are generated and divided into training (60%), validation (20%), and testing (20%) sets. Data normalization is applied to improve training stability and prevent overfitting.

The digital twin framework plays a significant role by simulating realistic TSCH network behavior and generating training data for offline learning. This enables offline training of the DNN, which can then be used to predict near-optimal scheduling decisions for new network conditions with significantly reduced computational complexity.

Fig. 6 illustrates the input–output relationship of the proposed deep neural network. The input consists of bipartite edge weights $W_{m,c}(n)$ while the output represents the corresponding optimal cell assignments c^*. The network learns to approximate the mapping between input features and optimal scheduling decisions obtained from the Hungarian algorithm.

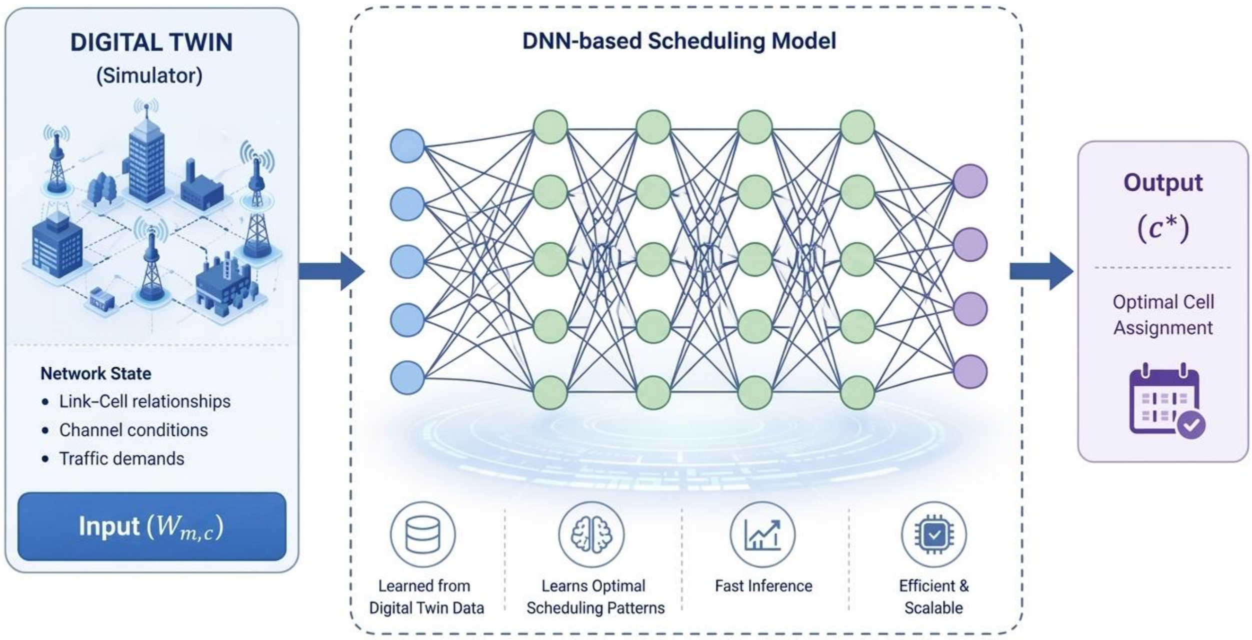

Fig 5: Input–output mapping of the digital twin-enabled DNN scheduling model.

5. PERFORMANCE EVALUATION

The effectiveness of the proposed scheduling framework is evaluated through comparative simulation analysis. The performance of the deep neural network (DNN) is compared with the conventional Hungarian (HG) scheduling method. The dataset is generated using the digital twin framework, and a total of 10,000 samples are used for training and evaluation.

The TSCH network simulation is implemented in MATLAB, while the DNN model is trained using Python with Keras and NumPy libraries. The dataset is divided into training (60%), validation (20%), and testing (20%) sets.

5.1 Network and Model Configuration

The bipartite graph is constructed based on the TSCH network model, where links and cells are mapped for scheduling. The DNN input consists of bipartite edge weights, and the output corresponds to optimal cell assignments obtained from the Hungarian algorithm.

For the considered setup, the total number of input features is given by the product of the number of links and cells, while the output represents the assigned cells for each link. A multilayer neural network with four hidden layers is used, and ReLU activation is applied to improve learning efficiency. Data normalization is employed to enhance training stability and avoid overfitting.

5.2 Execution Time Analysis

The execution time results demonstrate that the proposed DNN-based scheduling scheme significantly reduces computational complexity compared to the conventional Hungarian algorithm. As shown in Table 1, the average execution time of the DNN model is 31.5 ms, whereas the Hungarian method requires approximately 200 ms for each scheduling decision. This corresponds to nearly 84.25% reduction in execution time. The reduction is mainly achieved because the trained DNN predicts scheduling assignments directly without performing iterative combinatorial optimization during runtime. Such computational efficiency is highly beneficial for real-time TSCH network management, particularly in large-scale Industrial IoT environments where scheduling decisions must be updated dynamically under varying channel conditions. The average execution time per sample is summarized in Table 1

Table 1. Execution Time Comparison

Method	Execution Time (ms)
DNN	31.5
Hungarian (HG)	200

The results show that the proposed method achieves a substantial reduction in computation time while maintaining high accuracy.

5.3 Accuracy Evaluation

The accuracy results shown in Table 2 indicate that the proposed DNN model achieves consistently high prediction accuracy under different weighting factors. The highest accuracy of 93% is obtained when the weighting factor is set to 0.5, which represents a balanced consideration between throughput and delay optimization. For lower and higher weighting factors, the accuracy slightly decreases to 92%, indicating that the model remains stable across different scheduling priorities. These results confirm that the proposed learning-based framework can effectively approximate the optimal Hungarian scheduling decisions under diverse network conditions. The overall accuracy is computed as:

$$Accuracy = \frac{Total\ Correct\ Assignments}{Total\ \ Assignments} * 100\%$$

Table 2. Accuracy of Proposed DNN Scheme

Weighting factor (α)	Accuracy (%)
0.1	92
0.5	93
0.9	92

The results indicate that the proposed model achieves consistently high accuracy across different configurations.

5.4 Comparative Scheduling Performance Analysis

This subsection presents a comparative analysis of the proposed DNN-based scheduling scheme and the conventional Hungarian (HG) method. The evaluation is performed for a weighting factor $\alpha = 0.5$.

Fig. 6 illustrates the scheduling assignments generated by the proposed DNN model and the Hungarian algorithm for different link indices. It can be observed that the predicted assignments closely follow the optimal Hungarian solutions for most links. Only a few minor deviations are observed, which are mainly caused by approximation errors inherent in the supervised learning process. Despite these small mismatches, the DNN successfully captures the overall scheduling pattern and maintains near-optimal performance.

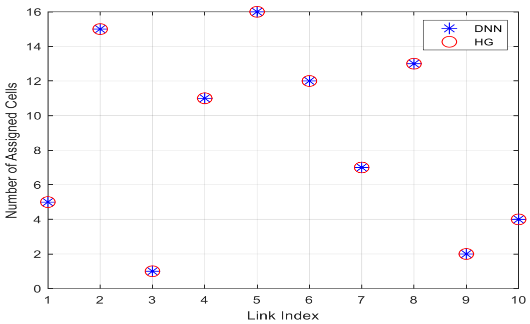

Fig 6: Comparison of cell scheduling results for all link indices using DNN and Hungarian (HG) methods ($\alpha = \mathbf{0.5}$)

Fig. 7 presents an example of scheduling mismatch between the DNN and Hungarian methods at link index 10. Although a slight deviation occurs in the selected cell assignment, the overall scheduling structure remains highly consistent with the optimal solution. The limited number of mismatches indicates that the proposed DNN model generalizes effectively across varying network conditions. Furthermore, the impact of such mismatches on overall throughput and delay performance is minimal.

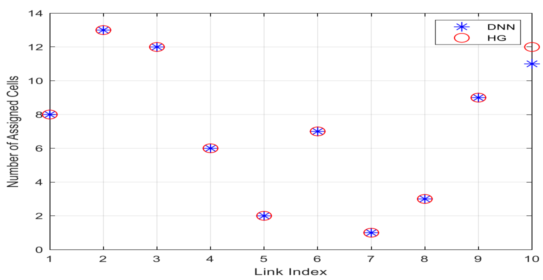

Fig 7: Example of cell scheduling comparison highlighting mismatch at link index 10 between DNN and Hungarian (HG) methods ($\alpha = \mathbf{0.5}$)

Fig. 8 compares the average throughput achieved by each communication link using the DNN and Hungarian scheduling methods. The results demonstrate that the proposed DNN model achieves throughput performance very close to the optimal Hungarian solution across all links. The throughput curves exhibit similar behavior, confirming that the learning-based model successfully preserves network efficiency while significantly reducing computational complexity. This validates the capability of the proposed framework for practical TSCH scheduling applications.

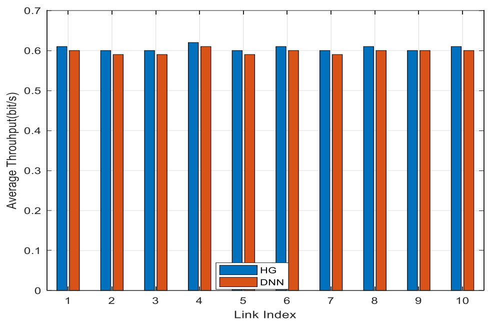

Fig 8: Average throughput of each link for DNN and Hungarian (HG) methods($\alpha = 0.5$)

5.5 Impact of Network Size on Scheduling Performance

To evaluate scalability, the proposed scheduling framework was tested under different network sizes demonstrate in Table3. The results indicate that the execution time of the Hungarian algorithm increases significantly as the number of nodes grows due to its combinatorial optimization complexity. In contrast, the DNN-based approach maintains relatively stable execution time while preserving high scheduling accuracy. This demonstrates the scalability and suitability of the proposed framework for large-scale Industrial IoT deployments.

Table 3. Scalability Analysis of the Proposed Scheduling Framework

Number of Nodes	DNN Accuracy	HG Execution Time	DNN Execution Time
10	94%	80 ms	15 ms
30	93%	140 ms	25 ms
50	92%	200 ms	31 ms
100	91%	350 ms	40 ms

6. CONCLUSION

This paper presented a digital twin (DT)-driven deep learning-based scheduling framework for IEEE 802.15.4e TSCH networks. The proposed approach modeled the scheduling problem as a weighted bipartite matching problem, where link-to-cell assignments were optimized using the Hungarian algorithm within the digital twin environment. The generated optimal scheduling data were then utilized to train a deep neural network (DNN) capable of predicting near-optimal scheduling decisions with significantly lower computational complexity. Simulation results demonstrated that the proposed DNN-based scheduling framework achieved performance comparable to the conventional Hungarian method while substantially reducing execution time. The obtained results confirmed that the proposed model can effectively maintain high scheduling accuracy and throughput performance under varying network conditions. Furthermore, the integration of digital twin technology enabled realistic data generation and efficient offline learning for intelligent TSCH network management. The proposed framework provides a scalable and computationally efficient solution for Industrial Internet of Things (IIoT) applications requiring reliable and low-latency communication. In addition, the learning-based scheduling mechanism has the potential to support real-time adaptive scheduling in large-scale wireless sensor and industrial automation networks. Future research may focus on extending the proposed framework to dynamic and large-scale network environments with varying traffic patterns and mobility conditions. Additional investigations can be conducted by incorporating reinforcement learning, federated learning, and edge intelligence to further enhance adaptive scheduling capability. Moreover, future work may include real-time hardware implementation, integration with 6G-enabled digital twin networks, and energy-aware scheduling optimization for sustainable Industrial IoT systems.

7. ACKNOWLEDGMENTS

The author gratefully acknowledges Mr. Orpon Chanda, Lab Engineer, Department of Electrical and Electronic Engineering (EEE), Leading University (LU), for his valuable support in formatting the references.

8. REFERENCES

[1] Ud Din et al., “The Internet of Things: A review of enabled technologies,” IEEE Access, 2019.

[2] E. Sisinni et al., “Industrial Internet of Things: Challenges and opportunities,” IEEE Trans. Ind. Informatics, 2018.

[3] B. H. Bae and S. H. Chung, “Fast synchronization in IEEE 802.15.4 TSCH,” Sensors, 2020.Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[4] T. Watteyne et al., “Reliability through frequency diversity,” PE-WASUN, 2009.

[5] S. Oh et al., “Autonomous scheduling scheme for TSCH,” Sensors, 2018.

[6] M. Ojo and S. Giordano, “Centralized scheduling in TSCH networks,” IEEE CSCN, 2016.

[7] N. Taheri Javan et al., “TSCH scheduling optimization,” IEEE Sensors Journal, 2020.

[8] S. Kharb and A. Singhrova, “Survey on TSCH scheduling,” J. Netw. Comput. Appl., 2019.

[9] G. Daneels et al., “Low-latency TSCH scheduling,” IEEE BlackSeaCom, 2019.

[10] M. Schluse et al., “Digital twins for Industry 4.0,” IEEE TII, 2018.

[11] T. Uhlemann et al., “The digital twin concept,” Procedia CIRP, 2017.

[12] F. Tao et al., “Digital twin in industry,” IEEE TII, 2019.

[13] R. Dong et al., “Digital twin for wireless systems,” arXiv, 2019.

[14] Y. Lu et al., “Digital twin edge networks,” IEEE TII, 2020.

[15] W. Sun et al., “Digital twin in 6G networks,” IEEE TVT, 2020.

[16] W. Sun et al., “Adaptive digital twin for IIoT,” IEEE TII,

2020.

[17] N. Samuel et al., “Learning to detect,” IEEE TSP, 2019.

[18] T. Wang et al., “Deep learning for wireless communications,” China Communications, 2017.

[19] H. Sun et al., “Learning to optimize,” arXiv, 2017.

[20] I. Hameed et al., “Deep neural networks for wireless optimization,” Applied Sciences, 2020.

[21] K. Date and R. Nagi, “Hungarian algorithm acceleration,” Parallel Computing, 2016.

[22] H. W. Kuhn, “The Hungarian method,” Naval Research Logistics, 2005.