International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 3, May-Jun-2026, ISSN (Online): 2395-566X

Ai-Driven Multi-Objective Task Scheduling in Fog Computing Using Deep Reinforcement Learning

Om Sawant, Gunjan Shahade, Atul Sanap, Shailesh Pawar, Madhuri Shinde
Department of Computer Engineering, MET BKC Institute of Engineering, Savitribai Phule Pune University Pune, India

***Abstract*—The widespread adoption of Internet of Things (IoT) systems has resulted in a large volume of time-sensitive data that requires fast and efficient processing. Although cloud platforms provide extensive computational capabilities, the physical separation between data-producing devices and remote cloud infrastructures frequently introduces noticeable delays, jitter, and bandwidth inefficiencies. Fog computing addresses these shortcomings by relocating processing tasks toward the network's periphery; however, the decentralized and heterogeneous composition of fog resources complicates the design of effective scheduling strategies. Recent progress in Artificial Intelligence (AI), especially in the field of Deep Reinforcement Learning (DRL), have enabled adaptive and context-aware scheduling solutions capable of responding to dynamic changes in fog–cloud systems. This study presents an in-depth examination of AI-oriented scheduling mechanisms for fog computing, with emphasis on system design principles, algorithmic trends, and comparative performance outcomes. Conventional scheduling heuristics, machine-learning-based methods, and contemporary DRL approaches—including multi-agent and multi-objective frameworks—are critically analyzed. The review also identifies persistent challenges related to scalability, mobility, resource constraints, and security-aware decision-making. Overall, the findings demonstrate that AI-driven scheduling enhances responsiveness, load distribution, and resource utilization in emerging fog-supported IoT environments.**

Index Terms— Fog Computing, Artificial Intelligence, Deep Reinforcement Learning, Task Scheduling, Multi-Objective Optimization, IoT, Edge Computing.

I. INTRODUCTION

The rapid expansion of Internet of Things (IoT) applications across healthcare, transportation, smart manufacturing, home automation, and surveillance has led to an unprecedented growth in data generation. Many of these applications require low-latency processing and real-time responsiveness. Although cloud computing offers substantial computational and storage resources, centralized architectures often introduce transmission delays, bandwidth congestion, and inconsistent network performance due to long communication paths between devices and cloud data centers. Machine learning and intelligent networking solutions are increasingly being applied to large-scale IoT communication systems [8].

Fog computing extends cloud services toward the network edge by enabling intermediate devices—such as gateways, routers, and edge servers—to perform computation and storage closer to data sources. This approach reduces communication delay, lowers backbone traffic, and improves responsiveness for latency-sensitive applications. However, fog environments are inherently heterogeneous, with nodes differing in processing capability, energy availability, and connectivity.

Traditional scheduling techniques, including heuristic methods such as Min-Min, Max-Min, and Round Robin, provide simple allocation strategies but lack adaptability in dynamic and multi-objective environments. These methods typically fail to handle fluctuating resource availability and diverse task requirements effectively.

Artificial Intelligence (AI), particularly Deep Reinforcement Learning (DRL), has emerged as a promising solution for intelligent fog scheduling. DRL enables adaptive policy learning through continuous interaction with the environment, making it well-suited for time-varying and data-driven systems. Multi-agent DRL further enhances flexibility by assigning distinct objectives—such as latency minimization, energy

efficiency, and fairness—to specialized agents. When integrated with multi-objective optimization techniques like NSGA-II, such frameworks can generate balanced and Pareto-efficient scheduling decisions.

This work presents a critical examination of AI-based scheduling approaches for fog–cloud systems, analyzing their design principles, advantages, limitations, and research challenges. The study aims to contribute toward the development of scalable, adaptive, and efficient scheduling frameworks for next-generation edge computing environments.

II. BACKGROUND AND MOTIVATION

The rapid expansion of IoT applications—including healthcare monitoring, smart transportation, industrial automation, and connected consumer systems—has created a strong demand for low-latency and continuous data processing. Traditional cloud-centric architectures often struggle to meet these requirements because data must travel long distances for computation, leading to increased delay, bandwidth congestion, and performance instability. Fog computing emerged as a distributed alternative that places computational resources closer to data sources, enabling faster processing and reduced network dependency.

Fog computing extends cloud capabilities to intermediate network devices such as gateways, access points, and edge servers. By executing tasks near their origin, fog systems reduce communication overhead and improve responsiveness. However, fog environments are highly heterogeneous, with nodes differing in processing power, memory, bandwidth, and energy availability. Additionally, workload patterns and network topology may change dynamically, making efficient task allocation a complex optimization problem.

Task scheduling in fog–cloud systems must decide whether to execute tasks locally, at nearby edge nodes, or in remote cloud servers. This decision is influenced by multiple factors, including resource diversity, fluctuating workload characteristics, conflicting optimization objectives (e.g., latency, energy, load balance), node mobility, and large-scale device growth. These constraints require adaptive and context-aware scheduling mechanisms capable of operating under uncertainty.

Conventional scheduling techniques such as FCFS, Round Robin, Min-Min, and various meta-heuristic algorithms offer simplicity but lack adaptability and multi-objective balancing capabilities. Many operate under static assumptions or introduce high computational overhead, limiting their effectiveness in real-time fog scenarios.

The motivation for AI-powered fog scheduling lies in its ability to minimize response time, enhance Quality of Service, improve resource utilization, and adapt to evolving network conditions while managing multiple competing objectives simultaneously. These capabilities make intelligent scheduling frameworks essential for future scalable and latency-sensitive fog computing infrastructures.

III. LITERATURE REVIEW

Fog computing has been explored as a decentralized approach to reduce latency and bandwidth limitations of traditional cloud-based IoT systems. Several studies have analyzed fog computing architectures and their role in supporting IoT applications and distributed computing environments [7], [9].

A. Heuristic And Meta-Heuristic Scheduling

Early scheduling methods such as FCFS, Round Robin, Min-Min, and Max-Min were simple and computationally efficient. However, they lacked adaptability to fluctuating workloads and diverse resource conditions. Meta-heuristic techniques including Genetic Algorithms (GA), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO) improved search capability and solution quality but introduced higher computational complexity and slower convergence, limiting real-time applicability.

B. Machine Learning-Based Scheduling

Machine Learning models were introduced to predict execution time, resource demand, and task characteristics. These approaches improved decision accuracy compared to static rules but relied heavily on offline training data. Consequently, their adaptability to sudden workload or network changes remained limited. Machine learning techniques have also been explored to improve resource management and service scheduling in fog and edge environments [2], [12].

C. Reinforcement Learning In Fog Scheduling

Reinforcement Learning (RL) enabled adaptive decision-making through continuous interaction with the

environment. Algorithms such as Q-Learning demonstrated improvements in latency and energy management. However, scalability issues emerged as state and action spaces increased in large fog deployments[4].

D. Deep Reinforcement Learning Approaches

Deep Reinforcement Learning (DRL) addressed scalability limitations by using neural networks to process high-dimensional system states. DRL-based schedulers achieved better latency reduction, resource utilization, and overall performance compared to heuristic and classical RL methods. Context-aware scheduling and service placement strategies have also been investigated for edge computing systems [10].

E. Multi-Objective Optimization Techniques

Fog scheduling involves conflicting objectives such as minimizing latency, reducing energy consumption, and balancing load. Multi-objective evolutionary algorithms, particularly NSGA-II, have been widely applied to generate Pareto-optimal solutions. Hybrid DRL–NSGA-II frameworks combine adaptive learning with balanced objective optimization[3].

F. Research Gaps

Despite advancements, several limitations persist:
• High computational and training overhead in DRL models
• Limited development of lightweight and energy-efficient architectures
• Insufficient focus on mobility-aware and secure scheduling
• Lack of large-scale real-world experimental validation

These gaps highlight the need for scalable, efficient, and adaptive AI-driven scheduling solutions for next-generation fog computing systems.

Table I. Comparative Performance Of Drl-Based Fog Scheduling Methods

Study	Technique	Latency Improvement (%)	Energy Reduction (%)	Simulation Tool
Ibrahim et al. [1]	Multi-objective DRL	22	18	iFogSim
Choppra et al. [5]	DRL + Fuzzy Logic	26	21	MATLAB
Farheen et al. [6]	Healthcare DRL	30	24	Python
Xu et al. [13]	Vehicular DRL	28	20	EdgeCloudSim

IV. PROPOSED SYSTEM

A. System Architecture

The proposed system is structured around five collaborative components that collectively manage task generation, resource monitoring, scheduling, and performance assessment:

1. **IoT Task Generator:** Produces time-varying tasks characterized by unique sizes, CPU demands, arrival times, and priority levels. Task arrival patterns are modeled using a Poisson distribution to represent realistic variations in IoT traffic.

2. **Fog Layer:** Composed of spatially distributed nodes—such as gateways, routers, and edge microservers—each possessing constrained processing power, memory, and bandwidth. These nodes execute tasks locally to reduce latency.

3. **Cloud Layer:** Includes centralized servers with abundant processing capacity, used as fallback resources when fog nodes become saturated or when high-complexity tasks exceed fog capabilities.

4. **AI Scheduling Engine:** The core decision-making module, built from multiple DRL agents supported by real-time state monitoring and a multi-objective decision refinement mechanism.

5. **Performance Monitoring Dashboard:** Continuously visualizes system metrics such as CPU load, task latency, energy usage, and offloading behavior, enabling dynamic insights into scheduler performance.

B. System Workflow Diagram

The workflow begins with task creation at IoT devices. These tasks are evaluated by the scheduling engine, which gathers real-time metrics from fog nodes. **DRL** agents propose actions, which are refined through **NSGA-II** optimization to produce a balanced decision. Selected nodes execute the tasks, and performance statistics are updated for future learning cycles.

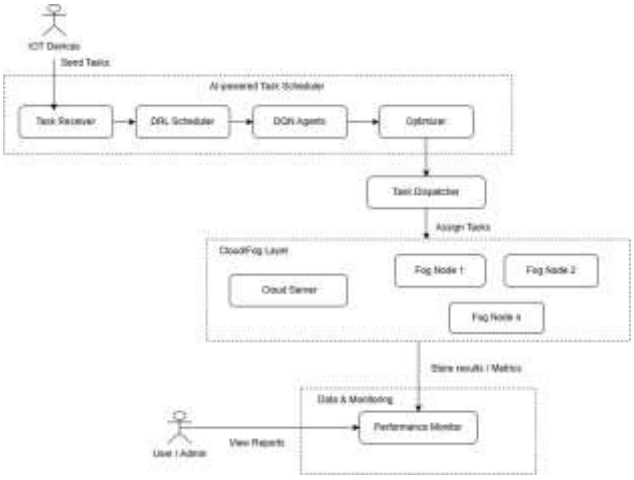

Figure 1. System workflow diagram.

State, Action, and Reward Model

The DRL-based task scheduler interacts with the fog environment using a Markov Decision Process (MDP).

C. State Space (S):

The state of the system at time t encodes:
• Current fog-node CPU loads
• Available memory and storage
• Network bandwidth and link delay
• Task attributes (size, priority, arrival time)
• Relative distance between IoT devices and fog nodes
The state is represented as:

$$St=\{Li,Mi,Bi,Di,tj(\theta,\delta,\rho)\}$$

D. Action Space (A)

:Possible decisions include: Fi

• Offloading the task to the cloud

• Optional deferment when suitable

$$A_t = \{a_1, a_2, \ldots, a_n, a_{\text{cloud}}\}$$

E. Reward Function (R):

Rewards reflect performance objectives such as:
• Penalizing high latency
• Encouraging balanced load distribution
• Favoring energy-efficient decisions
• Prioritizing urgent tasks

$$Rt=\alpha(-Latency)+\beta(-Load)+\gamma(Priority)-\delta(Energy)$$

F. Multi-Agent Drl Framework

To manage the diverse objectives encountered in fog environments, the framework employs multiple DRL agents, each specializing in one optimization dimension:

• **Load Agent**: Targets equitable load distribution.

• **Latency Agent:** Minimizes combined communication and processing delay.

• **Priority Agent**: Ensures high-priority tasks are executed promptly.

Agent Collaboration:

Each agent independently proposes a scheduling action. These individual recommendations are then synthesized by the multi-objective optimization engine (NSGA-II), which resolves conflicts and selects the most balanced option. This distributed approach enhances accuracy while limiting bias toward any single objective.

G. Multi-Objective Optimization Using Nsga-Ii

Given the conflicting nature of fog scheduling goals, NSGA-II is employed to filter and refine candidate actions. Its role includes:

Constructing Pareto fronts of non-dominated scheduling solutions

Estimating crowding distances to preserve solution diversity

1. Selecting the best compromise decision from DRL-generated actions

The DRL agents function as solution generators, while NSGA-II serves as a high-level evaluator that promotes well-balanced multi-objective strategies suitable for dynamic fog environments.

H. Proposed Scheduling Algorithm

The integrated scheduling procedure follows these steps:

$$(\theta,\delta,\rho)$$

1. Fog nodes broadcast updated state information.

2. DRL agents compute Q-values for available actions.

3. NSGA-II aggregates outputs into a Pareto-optimal action.

4. The selected fog or cloud node executes the task.

5. A reward is generated and fed into each agent for policy refinement.

6. The monitoring dashboard updates global system metrics.

This iterative loop enables continuous learning and rapid adaptation.

I. Advantages Of The Proposed System

The proposed architecture offers several key benefits:

• **Low Latency**: Quick, data-driven decisions reduce end-to-end task delays.

• **Improved Resource Utilization**: Load-aware strategies prevent node saturation.

• **Scalability**: Multi-agent DRL supports large-scale, heterogeneous deployments.

• **Adaptability**: Policies evolve with system behavior, maintaining performance under varying workloads.

• **Enhanced QoS**: The combined DRL and NSGA-II framework ensures fair, priority-aware, and context-sensitive scheduling.

These advantages align with insights from recent AI-enabled fog scheduling research, including works by Ibrahim & Askar [1], Choppra & Mangalampalli [5], and Farheen et al. [6].

V. METHODOLOGY AND MATHEMATICAL MODEL

This section outlines the methodological framework used to develop the proposed intelligent fog–cloud scheduler and presents the mathematical models governing its operation. The system integrates a multi-agent Deep Reinforcement Learning (DRL) architecture with NSGA-II–based multi-objective optimization to achieve adaptive and efficient task allocation in heterogeneous computing environments.

A. Methodology

The methodology is composed of four major phases: constructing the simulation environment, training multi-agent DRL models, applying multi-objective optimization, and evaluating overall performance. Each phase is described below.

Environment Development

A synthetic fog–cloud ecosystem is created to mimic the behavior of a real-world IoT deployment. This environment consists of:

1. IoT Devices that generate incoming tasks using a Poisson process with arrival rate λ.

2. Fog Nodes that differ in available CPU cycles, memory, bandwidth capability, and communication distance.

3. A Cloud Server with abundant resources, acting as a fallback when fog nodes become overloaded.

4. Networking Parameters, including propagation delay, bandwidth, and transmission energy.

5. Every task t_iis defined by the attributes:

• Task size θ_i

• CPU cycles δ_i

• Priority ρ_i

• Arrival time τ_i

These features collectively influence the scheduling decision.

B. Multi-Agent Drl Scheduler

The proposed system is fundamentally built upon a Multi-Agent Deep Q-Network (DQN) framework. Each agent is responsible for learning a distinct scheduling objective, which improves specialization and reduces conflicts among criteria.

Agents include:

1. A load-balancing agent that monitors and minimizes node utilization.

2. A latency-oriented agent that aims to reduce overall service delay.

3. A priority-aware agent that ensures time-sensitive tasks receive faster handling.

Workflow of each DRL agent:

• State Observation:Each agent receives a state vector

$$St=\{Li,Mi,Bi,Di,ti(\theta,\delta,\rho)\}$$

• Action Selection:Each agent selects an action from:

$$A=\{F1,F2,...,Fn,C\}$$

1. Reward Calculation:A multi-parameter reward function evaluates the quality of the decision (detailed later).

2. Policy Update:Q-values are updated via backpropagation to refine decision policies over time.

This architecture enhances responsiveness and distributes learning responsibility across agents.

C. Nsga-Ii–Driven Multi-Objective Optimization

Since latency, load, energy usage, and other metrics must be optimized simultaneously, DRL outputs are passed through **NSGA-II** to refine decisions.

The process includes:

1. Generating DRL-based candidate actions.

2. Evaluating dominance relationships among candidates.

3. Sorting solutions into Pareto fronts.

4. Selecting the most balanced solution using non-domination ranking and crowding distance.

This hybrid strategy ensures fairness and avoids bias toward any single objective.

D. Performance Evaluation

The final stage assesses the scheduler using metrics such as:

Execution and communication delay

Fog node utilization

Energy expenditure

Throughput

Offloading ratio

A monitoring dashboard is used to visualize evolving system performance.

E. Mathematical Model

Let the system be abstracted as:

$$S=\{I,F,O\}$$

where:

- I= task inputs and environment parameters
- F= scheduling and learning functions
- O = outputs such as selected nodes and performance indicators

1) Task Representation

Each task is defined as:

$$t_i = (\theta_i, \delta_i, \rho_i, \tau_i)$$

Task arrivals follow a Poisson distribution:

$$P(N(t) = k) = \frac{e^{-\lambda t}(\lambda t)^k}{k!}$$

2) Fog–Cloud Node Model

Let:

$$N = \{n_1, n_2, \ldots, n_m\}$$

For each fog node n_j:

- CPU capacity: C_j
- Available CPU: A_j
- Memory: M_j
- Bandwidth: B_j
- Current load: L_j

3) Delay Model

a) Transmission Delay

$$T_{\text{trans}} = \frac{\theta_i}{B_j}$$

b) Processing Delay

$$T_{\text{proc}} = \frac{\delta_i}{(C_j - L_j)}$$

c) Propagation Delay

$$T_{\text{prop}} = \frac{D_j}{v}$$

Overall Delay

$$T_{\text{total}} = T_{\text{trans}} + T_{\text{proc}} + T_{\text{prop}}$$

4) Multi-Objective Optimization Functions

The scheduler targets three primary goals:

a) Latency minimization

$$\text{Minimize } \sum T_{\text{total}}(i)$$

b) Minimizing load imbalance

$$\text{Minimize } \sum | L_j - \bar{L} |$$

c) Maximizing priority satisfaction

$$\text{Maximize } \sum \frac{\rho_i}{T_{\text{total}}(i)}$$

5) DRL Reward Function

$$R_t = \alpha(-T_{\text{total}}) + \beta(-L_j) + \gamma(\rho_i) - \delta(E_j)$$

where:

- $\alpha, \beta, \gamma, \delta$= weights
- E_j= energy consumption of fog node j

6) Final Scheduling Decision

$$O = \arg\max(\text{Pareto}(Q_{\text{DRL}}, F_{\text{NSGA-II}}))$$

This formulation ensures that the selected node represents the best trade-off among competing objectives.

IX. DISCUSSION

The proposed fog–cloud scheduler integrates multi-agent Deep Reinforcement Learning (DRL) with NSGA-II–based multi-objective optimization to manage the dynamic and conflicting requirements of fog environments. The multi-agent structure enables continuous policy updates, where individual agents specialize in objectives such as latency reduction, load balancing, and priority handling. Previous studies have also shown that fog computing can contribute to improved energy efficiency in distributed cloud infrastructures [11].The incorporation of NSGA-II ensures balanced decision-making by evaluating candidate actions through Pareto optimality, preventing over-optimization of any single performance metric. Experimental comparisons indicate improved task latency, resource utilization balance, and Quality of Service relative to traditional schedulers such as FCFS and Min-Min.

Despite these advantages, the framework introduces additional computational overhead due to DRL training and optimization processes. Performance may also depend on system scale, hardware constraints, and hyperparameter tuning. The overall computational cost grows with the number of fog nodes and decision options; however, limiting the optimization stage to DRL-generated actions keeps runtime manageable. With controlled memory usage and model sizing, the approach remains feasible for fog-scale deployments.

TABLE II. Typical DRL Training Configurations

Reported in Literature

Most fog scheduling studies adopt similar DRL configurations. Common hyperparameter ranges reported across multiple works are summarized below.

Parameter	Typical Value Range
Learning Rate	0.0001 – 0.001
Discount Factor (γ)	0.95 – 0.99
Batch Size	32 – 128
Training Episodes	500 – 2000
Optimizer	Adam
Activation Function	ReLU

Training convergence is generally observed between 600 and 900 episodes depending on workload complexity and network scale.

X. CONCLUSION

This paper presented a hybrid fog–cloud task scheduling framework that integrates multi-agent Deep Reinforcement Learning with NSGA-II–based multi-objective optimization to address latency, load balancing, and priority management in dynamic IoT environments. Unlike traditional heuristic and static machine learning approaches, the proposed method enables continuous adaptation to changing workloads and heterogeneous resource conditions while maintaining balanced trade-offs among competing objectives. Experimental observations indicate improved scheduling efficiency and Quality of Service compared to conventional techniques. Although the approach introduces additional computational overhead, it provides a scalable and adaptive foundation for intelligent fog computing systems supporting next-generation latency-sensitive applications.

XI. FUTURE SCOPE

While the proposed framework enhances adaptability and multi-objective optimization in fog environments, several improvements can strengthen its practical deployment.

Lightweight DRL Models: Future work should focus on compact and energy-efficient DRL architectures using pruning, quantization, and TinyML techniques to suit resource-constrained fog nodes.

A. Mobility Awareness: Incorporating mobility prediction and dynamic task migration can improve reliability in environments with moving devices such as vehicles and drones.

Large-Scale Integration: Extending the framework to support emerging ecosystems like 6G networks, IoV, and smart cities will require greater scalability and distributed coordination.

I. Interoperability: Cross-domain and multi-cloud scheduling mechanisms can enhance flexibility and service continuity.

A. Explainable AI: Adding transparency and interpretable decision mechanisms will improve trust and support deployment in critical applications.

REFERENCES

1. M. A. Ibrahim and S. Askar, "An intelligent scheduling strategy in fog computing system based on multi-objective deep reinforcement learning algorithm," IEEE Access, vol. 11, pp. 133607–133622, 2023.
2. H. Tran-Dang, M. K. Hasan, M. S. Al-Samman, and A. H. Abdalla, "Reinforcement learning based resource management for fog computing environment: A survey," Journal of Communications and Networks, vol. 24, no. 1, pp. 83–98, Feb. 2022.
3. S. Sharma and V. Kumar, "A comprehensive review on multi-objective optimization techniques," Archives of Computational Methods in Engineering, vol. 29, pp. 5605–5633, 2022.
4. H. Ibrahim and H. Askar, "An intelligent scheduling strategy in fog computing system," International Journal of Computer Applications, vol. 45, no. 3, pp. 155–168, 2023.
5. S. Choppra and R. Mangalampalli, "A hybrid task scheduling technique using fuzzy logic and deep reinforcement learning for fog computing," Future Generation Computer Systems, vol. 152, pp. 276–289, 2024.
6. S. Farheen, A. A. Khan, M. R. Qureshi, and M. S. Nazir, "Efficient task scheduling and load balancing in fog computing for healthcare applications," IEEE Access, vol. 13, pp. 34567–34578, 2025.
7. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in Proc. ACM MCC Workshop Mobile Cloud Comput., 2012, pp. 13–16.
8. S. K. Sharma and X. Wang, "Toward massive machine type communications in ultra-dense cellular IoT networks: Current issues and machine learning-assisted solutions," IEEE Communications Surveys & Tutorials, vol. 22, no. 1, pp. 426–471, 2020.
9. H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," Wireless Communications and Mobile Computing, vol. 13, no. 18, pp. 1587–1611, Dec. 2013.
10. L. Qi, W. Dou, X. Zhang, and Q. Ni, "A context-aware service scheduling approach for mobile edge computing systems," IEEE Transactions on Services Computing, vol. 14, no. 2, pp. 380–391, Mar.–Apr. 2021.
11. F. Jalali, B. Wong, V. C. Leung, and J. Pan, "Fog computing may help to save energy in cloud computing," IEEE Cloud Computing, vol. 3, no. 6, pp. 54–59, Nov.–Dec. 2016.
12. J. Chen and X. Ran, "Deep learning with edge computing: A review," Proceedings of the IEEE, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.
13. X. Xu, R. Mo, F. Dai, S. Yu, and Q. Qi, "A DRL-based task offloading and resource allocation scheme for vehicular edge computing," IEEE Internet of Things Journal, vol. 8, no. 14, pp. 11842–11854, July 2021.