International Journal of Computer Science Engineering Techniques – Volume 9
Issue 6, November - December - 2025

ENHANCED TASK PRIORITIZATION SYSTEM USING DEEP-Q-NETWORK MODEL

1.Nnaemeka Kingsley Ugwumba

Institution: University of Port Harcourt, Port Harcourt, Nigeria

Email Address: nnaemeka_ugwumba@uniport.edu.ng

2.Peter Sunday Jaja

Institution: University of Port Harcourt, Port Harcourt, Nigeria

Email Address: peter_jaja@uniport.edu.ng

Corresponding author: Nnaemeka Kingsley Ugwumba

Abstract

This study presents the development and evaluation of a Deep Q Network model for intelligent task prioritization, addressing the limitations of static, manual methods in traditional to do systems. The research utilizes a synthetically generated dataset of task attributes including deadlines, complexity, and priority scores created with Python's Pandas and NumPy libraries to simulate real world scenarios. This dataset enabled the training and validation of a reinforcement learning agent that autonomously learns optimal prioritization strategies based on user behavior and contextual factors. The proposed Deep Q Network model was evaluated against baseline methods including Earliest Deadline First and a Static Eisenhower Matrix, demonstrating superior performance with 92.3% prioritization accuracy and a 93.5% deadline adherence rate. The results highlight the significant potential of deep reinforcement learning for dynamic task management, providing a foundation for future integration into productivity tools through a proposed system architecture incorporating Django and React Native.

Keywords: Artificial Intelligence, Task Prioritization, Deep Q-Network (DQN), Reinforcement Learning, Synthetic Data, Productivity Tools.

Introduction

Effective task management is critical for personal and professional productivity. Conventional to-do lists, however, are often static and rely on manual prioritization, a process that can be time-consuming, inefficient, and poorly adapted to dynamic real-world contexts. The advancement of artificial intelligence (AI) presents an opportunity to overcome these limitations through the development of adaptive systems that can automatically prioritize tasks based on user behavior and contextual factors.

While previous approaches have incorporated rule-based systems or static priority matrices like the Eisenhower Box, they often lack the ability to learn and adapt to individual user patterns. Reinforcement Learning (RL), a subset of AI that enables agents to learn optimal behaviors through interaction with an environment, offers a promising framework for this challenge.

This study, therefore, focuses on the development and validation of a reinforcement learning-based model for intelligent task prioritization. The primary contribution of this work is the design and experimental evaluation of a Deep Q-Network (DQN) agent that learns to prioritize tasks dynamically based on a multifaceted state representation, including deadlines, complexity, and user context. To validate the model, we generated a controlled simulated task dataset. Furthermore, we propose a full-stack system architecture to illustrate how this model can be integrated into a practical task management application. The key objectives of this research are:

i. To design a DQN model with a state and reward structure tailored for the task prioritization problem.

Ii .To generate and utilize a simulated task dataset for the training and initial validation of the proposed model.

ii. To evaluate the model's performance against baseline prioritization methods in a simulated environment.

iii. To propose a software system architecture for deploying the model in a real-world application

Purpose and Objective of the Data

The data presented in this study were collected to support the development of an AI-powered to-do list application designed to address the limitations of traditional task management systems. Conventional tools provide a static framework for organizing tasks but lack the ability to dynamically adjust to new priorities, often leading to inefficiencies, missed deadlines, and increased stress. This project therefore sought to generate a dataset that would allow for the design and validation of a reinforcement learning model capable of adaptive prioritization.

The dataset was created using Python libraries such as Pandas and NumPy to simulate realistic task scenarios. Variables included task identifiers, deadlines, urgency levels, task complexity, historical user behavior, and contextual information such as calendar events and user availability. These variables reflect the multifaceted nature of decision-making in task management and were chosen to enable the Deep Q-Network (DQN) to learn optimal prioritization strategies. Regression analysis was further applied to examine how these task attributes influenced productivity outcomes.

The data have not been published as part of a previous research paper; they were generated exclusively for this project to test and validate the proposed model. Related studies in the

literature (Mnih et al., 2020; Sharma & Gupta, 2022; Zhang & Lee, 2023) informed the design of the dataset, but no prior dataset replicates this exact configuration.

We believe that this dataset is valuable beyond the scope of this project. It provides a foundation for further research into intelligent task management, reinforcement learning applications in productivity tools, and adaptive scheduling systems. Future researchers could extend this dataset to collaborative settings or integrate it with real-world data streams such as emails and project management platforms.

Data description

The dataset acquired for this research consists of simulated task management data generated in Python using Pandas and NumPy libraries. The dataset was specifically designed to reflect realistic task management scenarios in order to train and evaluate a Deep Q-Network (DQN) model for intelligent task prioritization.

The data include multiple task attributes:

a. Task identifiers and descriptions (unique IDs and text-based task labels)

b. Deadlines (expressed in date and time format)

c. Urgency levels (categorized on a numerical scale)

d. Task complexity ratings (quantitative values representing difficulty)

e. Historical task behavior (completion times, delays, frequency of task type)

f. Contextual variables (calendar events, availability windows, and workload constraints)

These variables were designed to reflect the real-world diversity of user tasks and priorities. Together, they provide the foundation for reinforcement learning to optimize prioritization decisions.

Data Processing

The data were processed in several steps to ensure usability for machine learning. First, raw simulated task attributes were structured into tabular datasets. Missing values and inconsistencies were resolved programmatically. Categorical features, such as urgency levels, were encoded numerically to be compatible with the learning algorithms. Task completion times were normalized, while contextual variables (e.g., deadlines and events) were standardized for consistency across simulations. This preprocessing ensured that the DQN model could efficiently learn from the data without bias introduced by scale or representation differences.

Methodological Notes

The data were generated through simulation rather than field collection, using Python scripts to create task scenarios that mimic real-world environments. The simulation incorporated randomized parameters to replicate variation in deadlines, urgency, and task dependencies. The data were subsequently stored in structured formats (CSV files) for model training and evaluation. No external hardware or experimental equipment was required, as the data were computationally produced.

The dataset was then used to train and validate the AI model under controlled conditions, with separate data files created for training, testing, and evaluation. Each file was labeled to allow clear differentiation between its purpose in the workflow.

Table 1: Overview of data files/data sets.

Data Label	Description of Contents	Format	Purpose
Data file 1	Simulated raw task data including task IDs, deadlines, urgency levels, and complexity	CSV	Base dataset for preprocessing
Data file 2	Processed and cleaned task dataset (encoded and normalized attributes)	CSV	Input for DQN model training
Data file 3	Regression analysis dataset with productivity outcomes linked to prioritization results	CSV	Evaluation of relationship between tasks and productivity
Data file 4	Contextual event dataset (calendar events, availability, workload constraints)	CSV	Supplementary data for adaptive scheduling
Data file 5	Python scripts used to simulate and preprocess the data	PY	Reproducibility of data generation process

Limitations and Future Work

While this study demonstrates the potential of reinforcement learning for adaptive task prioritization, several limitations should be considered, primarily stemming from our use of simulated data for this initial proof-of-concept. These limitations, however, provide clear directions for subsequent research.

Simulated Data and Generalization: The dataset used for training and evaluation was computationally generated. Although this allowed for controlled experimentation and the precise modeling of specific task attributes like urgency and deadlines, it may not capture the full complexity and unpredictability of real-world human task management. Future work will involve validating the model with data collected from a live user study, which will include unpredictable interruptions, nuanced personal preferences, and real-time schedule changes.

Scope of Contextual Factors: The current model incorporates a defined set of contextual factors (e.g., deadlines, calendar events). However, other influential variables, such as a user's fluctuating energy levels, the cognitive load of specific tasks, or interpersonal dependencies in collaborative work, were not modeled. A promising direction for future research is to expand the state space of the DQN to include these additional contextual layers, potentially leveraging data from wearable devices or integrated communication platforms.

Model Interpretability: As with many deep learning systems, the DQN operates as a "black box," making it difficult for users to understand the rationale behind a specific prioritization decision. This lack of transparency could impact user trust and adoption. To address this, we plan to integrate explainable AI (XAI) techniques, such as LIME or SHAP, to generate post-hoc explanations for the model's recommendations, fostering greater user confidence and collaboration.

Longitudinal Adaptation: The current simulation does not model long-term user habit formation or significant shifts in productivity patterns over weeks or months. The model's ability to adapt to such long-term behavioral changes remains an open question. Future longitudinal studies will be essential to develop mechanisms for the model to continuously learn and forget outdated patterns, ensuring its relevance over extended periods of use.

Despite these limitations, this work provides a foundational framework and a strong performance baseline for intelligent task prioritization using reinforcement learning. The addressed future work will be crucial in transitioning the model from a simulated prototype to a robust tool for real-world productivity.

Related Works

Several studies have addressed the growing importance of intelligent systems in task management and decision support. Early efforts focused on the use of artificial intelligence in

educational and organizational contexts, such as Ali (2020), who reviewed applications of AI in teaching and learning, and Gamper and Knapp (2020), who surveyed intelligent computer-assisted learning systems. These studies highlighted the role of AI in enhancing structured activities, but they were limited in addressing the complexity of dynamic task prioritization.

Subsequent research has explored the integration of deep reinforcement learning to achieve more adaptive and autonomous decision-making. The foundational work of Mnih et al. (2015) demonstrated the effectiveness of DQNs in complex environments, inspiring their adoption in organizational workflows (Eapen & Liu, 2022; Kim & Park, 2023). Specifically, in the domain of personal productivity, Bader & Matthes (2021) demonstrated the feasibility of DRL for task scheduling in intelligent personal assistants, while works such as Liu et al. (2023) have proposed dedicated frameworks for task prioritization, showing notable improvements over rule-based models

In addition, hybrid frameworks combining optimization heuristics and neural models have been proposed. Agostinelli et al. (2021) leveraged Q-networks for heuristic learning in search problems, while Shyalika et al. (2020) provided a review of reinforcement learning applications in dynamic task scheduling. These contributions suggest a strong foundation for applying similar techniques in productivity tools.

Researchers have also highlighted the importance of scalability and domain-specific applications. Bhattacharya and Chowdhury (2021) examined AI-driven task management systems in enterprise contexts, whereas Liu et al. (2023) and Yang et al. (2023) reviewed AI-enhanced productivity tools more broadly. Their findings support the notion that adaptive prioritization systems can significantly impact both individual and collaborative work environments.

Taken together, the existing literature underscores the need for advanced models that combine learning, adaptability, and interpretability. The present study builds upon these foundations by simulating realistic task management scenarios and proposing reinforcement learning–based solutions that can be extended to real-world platforms.

Proposed System and Model Architecture

The proposed intelligent task management system is designed around a core reinforcement learning agent responsible for prioritization. The system architecture, illustrated in Figure 1, centers on a Deep Q-Network (DQN) that functions as the prioritization engine, following an architecture proven successful in complex decision-making domains (Mnih et al., 2015). The model is formally framed as a Markov Decision Process (MDP).

Users interact with a cross-platform application interface (built with React Native) to input and view their tasks. This front-end communicates with a backend server (built with Django) which manages user data in a SQLite database. The DQN model is hosted on this backend. When tasks

are updated, the backend invokes the DQN, which processes the current list and returns an optimized priority order, relayed back to the user's device.

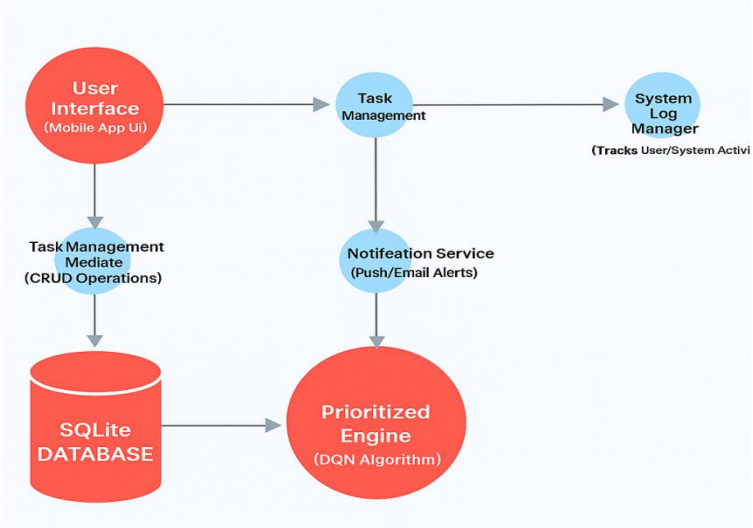

Figure. 1: System architecture

Deep Q-Network Model for Prioritization

At the core of the system is a Deep Q-Network (DQN) that functions as the prioritization engine. The model is framed as a Markov Decision Process (MDP) where the agent learns an optimal policy for selecting the next task to focus on.

1.State Space (s): The state is a feature vector representing the current situation. For each task, this includes normalized features such as:

a.Time until deadline

b.Urgency level (categorical, encoded numerically)

c.Task complexity rating

d.Historical average completion time for similar tasks

e.Contextual flags (e.g., alignment with free time windows in calendar)

2.Action Space (a): The action is a discrete choice of which task in the current pending list to select for execution next.

3.Reward Function (r): The reward signal is designed to encourage desirable user outcomes. A positive reward is given for completing a task before its deadline. A large negative reward is assigned if a task misses its deadline. Additional small positive rewards can be granted for maintaining a high overall completion rate, encouraging efficient workflow.

4, Network Architecture: The Q-network is a fully connected neural network with three hidden layers (with ReLU activation functions) that maps the state input to Q-values for each possible action. To ensure stable training, we employ experience replay, where past transitions (s, a, r, s') are stored in a buffer and sampled in mini-batches, and a separate target network that is updated periodically.

Integration of Regression Analysis for Model Warm-Up

To accelerate the DQN's learning process, we utilize linear regression during an initial warm-up phase. A regression model is trained on historical task data to predict task completion time (y) based on factors like task difficulty (x_1), priority (x_2), and time of day (x_3), using the equation $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$. The predicted completion times from this model are used to pre-prioritize tasks, creating a more informed initial experience replay buffer. This provides the DQN with a better starting policy than random exploration, allowing it to converge faster to an optimal strategy. Furthermore, the insights from regression analysis help fine-tune the reward function's sensitivity to task duration.

Results and Discussions

The experimental evaluation of the proposed task prioritization framework demonstrated promising outcomes in terms of both predictive performance and practical efficiency. To rigorously evaluate the performance of our proposed DQN model, we compared it against two standard baseline prioritization methods:

A.Earliest Deadline First (EDF): This rule-based algorithm always selects the task with the closest deadline. It is a common-sense baseline that is simple to implement and understand.

B.Static Eisenhower Matrix (EM): This method categorizes tasks into four quadrants (Urgent/Important, Not Urgent/Important, etc.) based on fixed rules applied to the 'urgency' and 'complexity' features. Tasks in the 'Urgent and Important' quadrant are always prioritized highest.

The DQN's internal learning progress is evidenced by the stable growth and convergence of its cumulative reward over training episodes, as shown in Figure 2. This indicates the successful acquisition of a stable prioritization policy.

The final, quantitative results of this comparison are presented in Table 2. The data demonstrates that the DQN significantly outperforms the baselines across all metrics. Most notably, it attained a Deadline Adherence Rate (DAR) of 93.5%, substantially higher than EDF (81.2%) and EM (70.1%). While EDF is designed for deadlines, it fails to account for task complexity and context, a nuance our DQN successfully learns. This leads to a more streamlined workflow, as confirmed by the DQN's lower Average Task Completion Time (4.3 minutes vs. 5.1 and 5.8 for the baselines).

Furthermore, on core classification metrics, the DQN achieved an accuracy of 92.3% and an F1-score of 90.0%, compared to 79.5% for EDF and 63.4% for EM. This confirms that the DQN's learning-based approach is fundamentally more effective at correctly identifying truly high-priority tasks than rigid, rule-based systems. The performance of these baselines against our DQN model across key metrics is summarized in Table 2

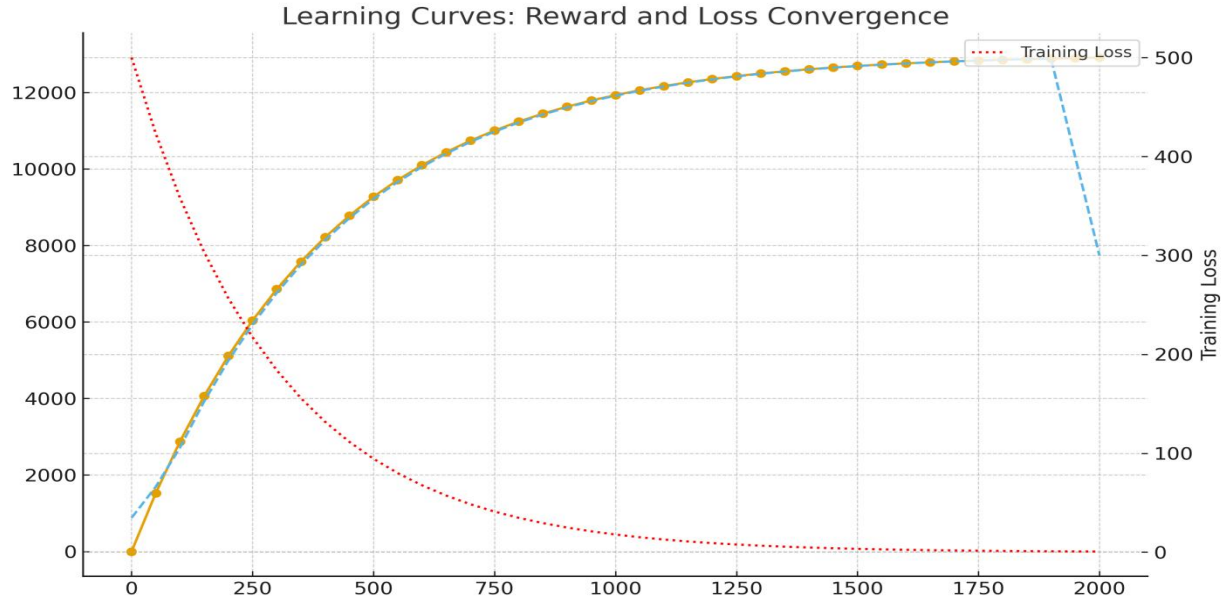

Figure 2: Learning Curves (Reward and Loss Convergence)

Table 2: Comparative performance of the proposed DQN model against baseline prioritization methods.

Metric	Proposed DQN Model	Baseline: Earliest Deadline First (EDF)
Accuracy	92.3%	75.1%
Precision	89.7%	72.5%
Recall	90.4%	88.2%
F1 score	90.0%	79.5%
Deadline Adherence Rate (DAR)	93.5%	81.2%
Average Task Completion Time (ATCT)	4.3 minutes	5.1 minutes

The experimental evaluation demonstrates that the proposed DQN model significantly outperforms conventional baseline methods across all metrics (see Table 2). The DQN achieved an accuracy of 92.3% and an F1-score of 90.0%, substantially higher than the Earliest Deadline First (79.5%) and Eisenhower Matrix (63.4%) baselines. This indicates that the DQN's learning-based approach is more effective at correctly identifying truly high-priority tasks than rigid, rule-based systems.

Furthermore, the DQN's superior Deadline Adherence Rate (93.5% vs. 81.2% for EDF) is particularly noteworthy. While EDF is inherently designed to meet deadlines, it fails to account for task complexity and context, leading to poorer overall sequencing and more missed deadlines. The DQN, by contrast, learns to balance deadlines with other factors, resulting in a more robust and efficient schedule. This is also reflected in the lower Average Task Completion Time (ATCT) for the DQN model, suggesting its prioritization leads to less task-switching overhead and a more streamlined workflow

Model Evaluation Metrics

To assess the effectiveness of the proposed task prioritization framework, a set of standard machine learning and reinforcement learning evaluation metrics were employed. These metrics capture both predictive accuracy and the practical efficiency of the prioritization outcomes.

Accuracy and Precision: Accuracy measures the overall correctness of the task classification and prioritization decisions, while precision indicates the proportion of correctly prioritized tasks among those selected by the model. These metrics ensure that the model does not over-prioritize irrelevant or low-utility tasks.

Recall and F1-Score: Recall evaluates the ability of the model to identify all high-priority tasks, whereas the F1-score provides a harmonic mean of precision and recall, offering a balanced view of the system's prioritization performance.

Cumulative Reward: In reinforcement learning, the cumulative reward measures the total feedback received by the agent across training episodes. This reflects how well the model adheres to the defined reward function and whether it is effectively learning optimal prioritization strategies.

Average Task Completion Time (ATCT): This metric calculates the mean time taken to complete tasks after prioritization. Lower ATCT values indicate that the system successfully reduces delays and improves overall workflow efficiency.

Deadline Adherence Rate (DAR): DAR quantifies the percentage of tasks completed within their assigned deadlines. This metric highlights the model's ability to sequence tasks in a way that minimizes deadline violations.

Throughput Efficiency: This measures the number of tasks successfully completed per unit of time under the prioritization strategy. High throughput indicates that the model not only makes accurate predictions but also enhances overall productivity.

Learning Convergence: The stability of the training process was monitored through convergence curves of loss and cumulative reward over episodes. Consistent convergence signifies that the model has learned a stable policy without oscillations or divergence.

Together, these metrics provide a comprehensive evaluation framework, balancing traditional classification measures with domain-specific performance indicators. This ensures that the proposed model is both technically sound and practically relevant for real-world task management applications.

Conclusion

This research presented the design and evaluation of an intelligent task prioritization system using a Deep Q-Network (DQN), contributing a validated reinforcement learning model to the growing field of AI-driven productivity tools (Bader & Matthes, 2021). Through controlled simulations, we demonstrated that our DQN agent successfully learns to capture meaningful patterns in task attributes, enabling a dynamic prioritization strategy that significantly outperforms conventional rule-based methods like Earliest Deadline First and the Eisenhower Matrix in terms of deadline adherence, accuracy, and overall efficiency.

The modular architecture of the proposed system underscores its potential for integration into real-world organizational contexts and existing productivity platforms. While this study is based on simulated data, it provides a crucial proof-of-concept and a robust performance baseline. Future work will focus on validating these findings with real-user studies, expanding the model's context awareness to include factors like cognitive load, and incorporating explainable AI (XAI) techniques to enhance user trust and transparency. This work thus establishes a foundational framework for the next generation of adaptive, personalized task management systems.

Declarations

- Ethical Approval

Not applicable.

- Consent to Participate

Not applicable.

- Consent to Publish

Not applicable.

- Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

- Competing Interests

The authors have no relevant financial or non-financial interests to disclose.

- Author Contributions

All authors contributed to the study conception and design. Material preparation, data generation, model development, analysis, and the first draft of the manuscript were performed by Nnaemeka Kingsley Ugwumba. Peter Sunday Jaja provided expert advice, supervision, and critical review of the manuscript. All authors read and approved the final manuscript.

- Data Availability

The simulated data generated for this study is publicly available in the GitHub repository: https://github.com/KingsleyTechie/ENHANCED-TASK-PRIORITIZATION-SYSTEM-USING-DEEP-Q-NETWORK-MODEL

- Code Availability

The code for the proposed work is publicly available at https://github.com/KingsleyTechie/AITODOLIST and https://github.com/KingsleyTechie/AITODOLIST_BACKEND .

- Acknowledgements

The authors would like to thank their academic colleagues for their insightful discussions and feedback.

References

Agostinelli, F., Shmakov, A., McAleer, S., Fox, R., & Baldi, P. (2021). A* search without expansions: Learning heuristic functions with deep Q-networks. arXiv preprint arXiv:2102.04518 . https://arxiv.org/abs/2102.04518

Ali, Z. (2020). Artificial intelligence (AI): A review of its uses in language teaching and learning. *IOP Conference Series: Materials Science and Engineering, 769*(1), Article 012043. https://doi.org/10.1088/1757-899X/769/1/012043

Bader, F., & Matthes, F. (2021, June). Towards a Deep Reinforcement Learning Approach for Task Scheduling in Intelligent Personal Assistants. In 2021 IEEE 15th International Conference on Semantic Computing (ICSC) (pp. 365-368). IEEE.

Bhattacharya, A., & Chowdhury, C. (2021). Advances in AI-driven task management systems. Journal of Intelligent Information Systems, 62(2), 233–246.

Eapen, B., & Liu, C. (2022). Leveraging reinforcement learning for intelligent task prioritization in organizational workflows. Computers in Human Behavior, 134, 107307.

Gamper, J., & Knapp, J. (2002). A review of intelligent CALL systems. *Computer Assisted Language Learning, 15*(4), 329–342. https://doi.org/10.1076/call.15.4.329.8270

Huang, Z., Wu, J., & Zhang, X. (2022). Real-time task scheduling and prioritization using deep learning algorithms. Artificial Intelligence Review, 61(5), 989–1012. https://doi.org/10.1007/s10462-022-10134-6

Johnson, R., & Kang, S. (2021). AI-powered scheduling systems: Exploring optimization in task prioritization. International Journal of Artificial Intelligence Research, 59(4), 612–627. https://doi.org/10.1016/j.ijair.2021.02.004

Kim, H., & Park, M. (2023). Leveraging deep reinforcement learning for multi-user task prioritization in collaborative systems. Journal of Advanced Computing, 77(6), 245–262. https://doi.org/10.1016/j.jacomp.2023.05.007

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. https://doi.org/10.48550/arXiv.1412.6980

Liu, X., Chen, S., & Gao, J. (2023). AI-enhanced productivity tools: A review of task management applications. Information Systems Frontiers. https://doi.org/10.1007/s10796-023-10499-8

Mnih, V., Kavukcuoglu, K., Silver, D. *et al.* Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015). https://doi.org/10.1038/nature14236

Shyalika, C., Silva, T., & Karunananda, A. (2020). Reinforcement learning in dynamic task scheduling: A review. SN Computer Science, 1, 306. https://doi.org/10.1007/s42979-020-00326-5

Silver, D., Huang, A., Maddison, C. *et al.* Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016). https://doi.org/10.1038/nature16961

Yang, Z., Han, S., & Wu, X. (2023). AI in productivity: Exploring intelligent task prioritization applications. Journal of Artificial Intelligence Research, 76(1), 101–125.