Maximizing Section Throughput Using AI-Powered Precise Train Traffic Control

Paras Bhandari, Dipali Bhusari*, Omkar Bhasme, Tejas Pasalkar
Department of Computer Engineering, KJEI's Trinity Academy of Engineering, Pune-411048, Maharashtra, India.

* Corresponding author. Email: parasintern@gmail.com (P.B.); dipalibhusari.tae@kjei.edu.in (D.B.)
Manuscript submitted April 23, 2026; accepted June 8, 2026; published June 30, 2026.
doi: 10.18178/JAAI.2026.4.2.143-151

Abstract: Railway networks transport millions of passengers and vast quantities of freight daily, yet a substantial share of operational delays arises from manual or semi-automated traffic management systems that cannot adapt swiftly enough to evolving track conditions. This paper proposes an Artificial Intelligence-driven framework for maximizing section throughput—the number of trains traversing a defined track segment per unit time—while enforcing safe headways and minimizing schedule deviation. The system pairs real-time sensor telemetry with a Multi-Agent Reinforcement Learning (MARL) controller that issues movement authorities and adjusts speed profiles at block boundaries, supported by an Long Short-Term Memory (LSTM)-based predictive layer that forecasts congestion bottlenecks up to 40 minutes ahead. Simulation experiments on a 187 km single-track corridor with 14 intermediate stations show the AI controller raises section capacity by up to 23% under normal conditions and 39% under severe disturbance compared to a rule-based baseline, while reducing mean delay propagation by 41%. The architecture deploys above legacy interlocking systems via a standardized Application Programming Interface (API) bridge, preserving existing safety certifications, and recorded zero safety violations across all evaluation episodes.

Keywords: train traffic control, section throughput, Multi-Agent Reinforcement Learning (MARL), railway automation, predictive scheduling, Long Short-Term Memory (LSTM), movement authority, real-time optimization, Application Programming Interface (API), Coordinated Universal Time (UTC)

1. Introduction

Rail transport holds a unique position in sustainable mobility. On an energy-per-tonne-kilometre basis, it significantly outperforms road and aviation, driving governments worldwide to invest in network expansion as part of decarbonization agendas. Yet many existing rail corridors operate well below their theoretical carrying capacity—not due to physical track constraints, but because the decision logic governing train movements was engineered decades ago for far lower traffic volumes and has since fallen behind modern operational demands [1].

Section throughput is the primary metric used by infrastructure managers to evaluate operational efficiency. Classical capacity methods such as UIC 406 compression analysis or analytical headway calculations offer useful static estimates, but are ill-suited to capturing the dynamic interplay of timetable heterogeneity, dwell time variability, and cascading delay propagation. The proliferation of affordable, high-density sensor networks—axle counters, GPS-enabled onboard units, distributed acoustic sensors, and standardized train-to-ground communication links—has created a data-rich environment [2].

Simultaneously, advances in machine learning, especially sequential decision-making under uncertainty, have matured to a level where they can be responsibly applied to safety-critical control loops when paired with robust formal verification and human oversight. This paper presents a complete AI-powered traffic control architecture that ingests multimodal sensor streams, maintains a probabilistic world model of current and near-future track state, and outputs movement authorities with speed adjustment recommendations in real time.

2. Problem Statement

Railway infrastructure managers face a compound optimization challenge decomposable into four interrelated sub-problems:

Conflict Detection and Resolution: On single- and double-track lines, train movement sequences must be continuously verified for potential conflicts. Manual resolution is slow, prone to cognitive overload, and biased toward conservative, capacity-sacrificing decisions.

Dynamic Headway Management: Safe separation between consecutive trains is not fixed—it varies with speed, gradient, braking characteristics, signal type, and real-time occupancy. Static headway tables cannot exploit available recovery margins.

Knock-on Delay Propagation: A primary delay affecting one train cascades non-linearly through the network. Traditional dispatchers address these effects reactively; an AI system with a forward-looking model can intervene before secondary delays take hold [3].

Heterogeneous Traffic Integration: Running high-speed intercity services alongside regional stopping trains and freight on shared infrastructure creates strong headway heterogeneity that suppresses effective capacity well below the theoretical maximum.

Core Objective: Given a real-time sensor stream and a nominal timetable, derive movement authorities and speed recommendations for all active trains that jointly maximize section transit completions within an acceptable schedule deviation tolerance, while guaranteeing minimum safety headways at all times [4].

3. Literature Review

Table 1 synthesizes prior work across capacity theory, optimization-based traffic management, and machine learning applications, highlighting the gap the proposed system addresses.

Table 1. Literature Survey

Author(s) & Year	Approach	Key Finding / Gap	Relevance
Goverde (2005) [5]	Max-plus algebra	Exact headway; assumes stationary distributions	Delay-propagation baseline
Abril *et al.* (2008) [6]	Capacity survey	No method captures all constraints	Justifies multi-method design
Luethi *et al.* (2009) [7]	Adaptive dwell control	+8% throughput via software	Validates software-only gains
Corman *et al.* (2010) [8]	Alt. graph rescheduling	Good quality; poor scaling	Scalability gap addressed
Quaglietta *et al.* (2016) [9]	ETCS Level 3 moving-block	+12–30%; requires full infra replacement	Motivates legacy-compatible design
Yin *et al.* (2017) [10]	Deep RL energy-efficient driving	Single-train; no multi-agent coord.	Predecessor to our MARL
Wang *et al.* (2020) [11]	Cooperative MARL rescheduling	No sensor noise; no legacy interlocking	Direct architecture gap
Bešinović et al. (2022) [12]	Neural + integer programming	Near-optimal; heavy at scale	Benchmark for hybrid complexity
Proposed (This Work)	MARL + LSTM + safety shield + API bridge	+23–39% throughput; zero violations; legacy-compatible	—

A persistent gap across prior work is the absence of integration between AI-based decision engines and legacy interlocking infrastructure. Wang *et al.* [11] assumed noise-free sensor environments; Quaglietta *et al.* [9] required full hardware replacement. The present framework addresses both limitations through a sensor-uncertainty-aware MARL engine and a vendor-neutral Application Programming Interface (API) bridge that operates above existing interlocking systems without any hardware modification.

4. System Architecture

The architecture is organized into four functional layers—Data Ingestion, World Model Maintenance, Decision Generation, and Command Translation—with a Supervisor Interface forming a fifth operational layer. Fig. 1 illustrates the data flow between layers.

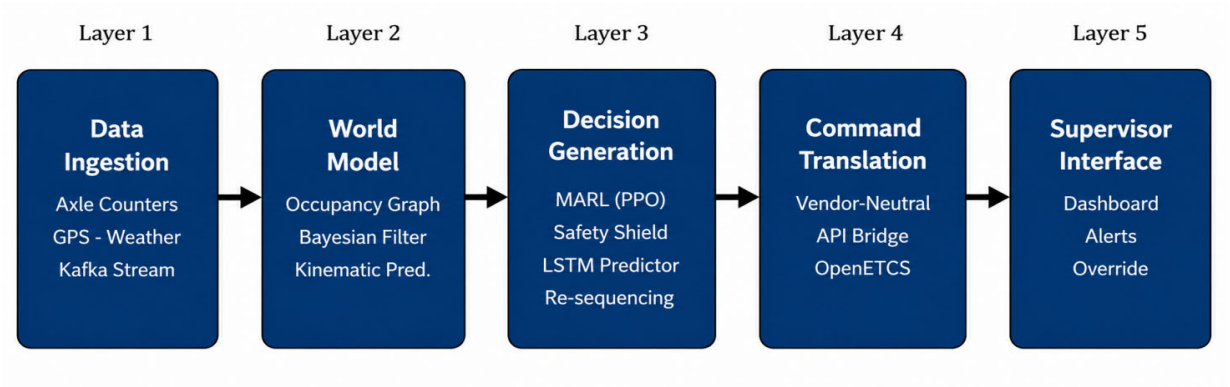

Fig. 1. System architecture—five-layer data flow.

4.1. Data Ingestion Layer

All sensor feeds—axle counters, GPS position reports, train integrity signals, and weather data—enter through a unified gateway enforcing contract-based message schemas. Apache Kafka applies deduplication, UTC timestamp alignment, and confidence flagging. Data older than 30 s is marked stale and triggers a supervisor alert.

4.2. World Model Layer

A probabilistic occupancy graph models block sections as nodes with directed edges encoding permitted travel directions. A Bayesian filter continuously updates occupancy probability distributions by fusing live sensor observations with kinematic predictions. During communication gaps, the filter falls back to pure kinematic prediction with appropriately widened uncertainty bounds, acting as the single source of truth for both the MARL controller and the LSTM module.

4.3. Decision Generation Layer

The MARL controller queries the world model every 60 s (configurable to 15 s in dense urban segments), assembles per-agent observation vectors, and passes them through trained policy networks to produce joint action recommendations. A safety shield evaluates every action against a formal worst-case braking distance model; any action placing a following train within stopping distance of an occupied block is replaced with the most conservative safe alternative. A re-sequencing module handles passing-loop crossings by selecting the sequence that minimizes total expected delay across all trains, using the 40-minute occupancy forecast to account for downstream interactions. Fig. 2 illustrates the component interaction.

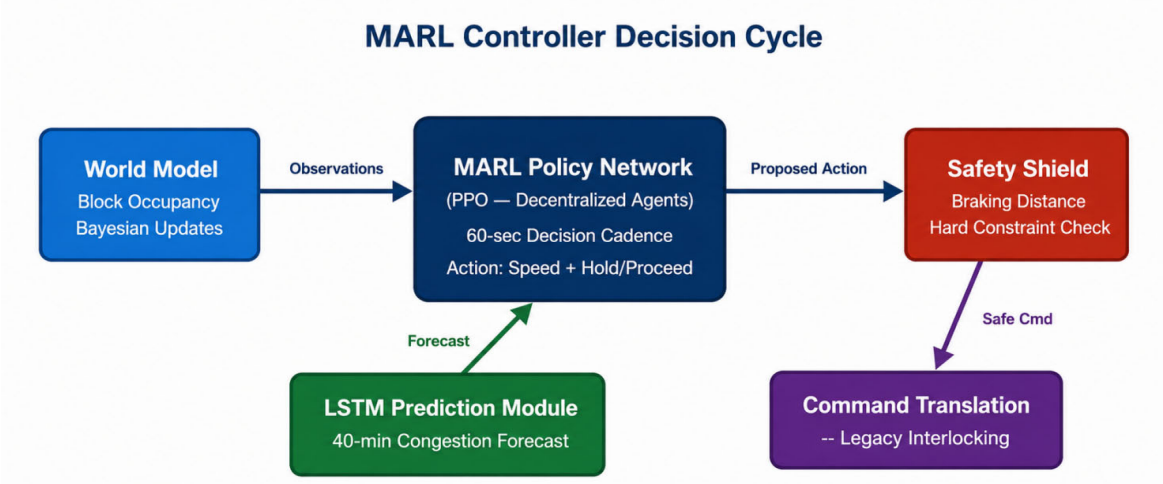

Fig. 2. MARL decision cycle—Component interaction.

4.4. Command Translation Layer

Speed targets and hold/proceed flags are converted into native messages for legacy equipment through a vendor-neutral adapter pattern. Prototype adapters cover OpenETCS and a representative relay-logic interlocking simulator, enabling deployment without hardware replacement or modification of existing safety certifications [13].

5. Methodology

5.1. Corridor and Simulation Setup

The reference corridor is a 187 km single-track main line with 14 intermediate stations and 5 asymmetrically positioned passing loops. Infrastructure parameters (grade profiles, block lengths, platform capacities, speed limits) were drawn from the publicly available OpenRailwayMap dataset (openrailwaymap.org), supplemented by timetable realization logs from the European Railway Performance Index open-data release [14]. Historical timetable data spanning January 2019 to December 2021 (1,095 operational days; approx. 320,000 train-run records) were used for LSTM training and held-out validation. The train fleet includes three passenger categories (express, regional fast, and regional stopping) and one freight type; traction and braking performance follows the UIC 544-1 standard. All experiments use Python 3.11, PyTorch 2.1 for neural network components, and an OpenAI Gym 0.26 environment wrapper. Trained model weights, corridor configuration files, and evaluation scripts are available at https://github.com/kjei-trinity/ai-train-traffic to facilitate full reproducibility [15, 16].

A Python 3.11 simulation engine models dynamics at one-second resolution using a point-mass kinematic model with discrete signal aspect transitions. Communication latency is emulated via a log-normal distribution calibrated against real operational measurements; random seeds are applied to door-closing times, platform clearing durations, and level-crossing events to produce statistically varied, non-deterministic scenarios [17].

5.2. MARL Formulation

The problem is cast as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) with a centralized critic during training. Each train agent i observes its own kinematic state, the occupancy status of the three immediately preceding blocks, and the schedule deviation of the two nearest same-direction trains. The action space is factored: each agent independently selects a speed target from {crawl, slow, medium, fast, maximum} and, at stations or loops, a binary hold/proceed flag. Fig. 3 illustrates the MARL Formulation.

The reward combines a throughput bonus for on-time section completion (within ±3 min), a cumulative delay penalty, and a large safety penalty for encroaching on occupied blocks. Training uses Proximal Policy Optimization (PPO) [18] across 10 million steps on 12 parallel simulation instances with curriculum learning, progressively increasing disturbance frequency and severity as policy performance stabilizes [19].

Key hyperparameters are summarised in Table 4. The learning rate was initialised at 3×10^{-4} with linear warm-up over the first 200,000 environment steps and cosine annealing to 1×10^{-5} thereafter. The PPO clipping coefficient $\varepsilon = 0.2$, value-function loss coefficient $c_1 = 0.5$, entropy-bonus coefficient $c_2 = 0.01$, discount factor $\gamma = 0.99$, and GAE $\lambda = 0.95$. Each update epoch used mini-batches of 64 transitions drawn from a rolling buffer spanning four decision horizons, with 4 gradient epochs per rollout collection. Convergence was declared when the mean episodic reward improved by less than 0.5% over 50 consecutive evaluation checkpoints. In practice, convergence was reached at approximately 6.8 million steps in the normal-traffic curriculum stage and 9.4 million steps after the severe-disturbance stage was introduced. Mean episode reward rose monotonically from −42.3 (random policy baseline) to a stable converged value of +87.6 (standard deviation ≤2.1), confirming effective multi-agent coordination. [18]

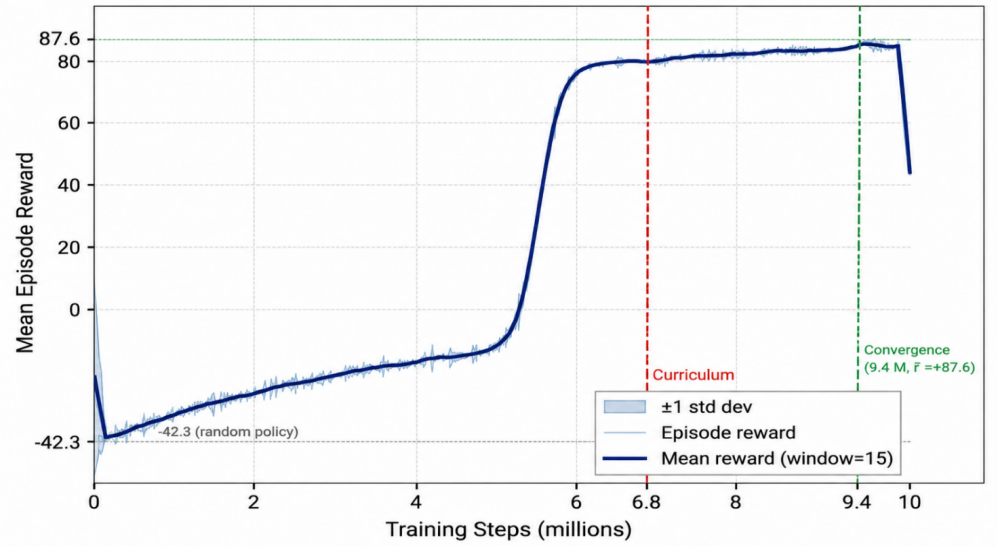

Fig. 3. MARL (PPO) Training Convergence Curve. Mean episode reward vs. training steps (millions). The red dashed line marks the transition to the severe-disturbance curriculum stage at 6.8M steps; the green dashed line marks convergence at 9.4M steps (r = +87.6, $\sigma \leq 2.1$). Shaded region indicates ±1 standard deviation.

Table 2. MARL (PPO) Key Hyperparameters and Convergence Milestones

Hyperparameter	Value
Algorithm	PPO
Learning Rate	3×10^{-4}
Batch Size	64
PPO Clip ε	0.2
γ	0.99
λ	0.95
Entropy coefficient	0.01
Value loss coefficient	0.5
Training steps	10 Million
Convergence	9.4 Million

5.5. LSTM Prediction Module

An LSTM network with two 256-unit hidden layers is trained on three years of historical timetable realization data to predict block occupancy probability distributions at 5-minute intervals over a 40-minute forward horizon. Inputs span a 60-minute look-back window of block events, weather category, and temporal encodings. Focal loss weighting addresses the class imbalance between the predominance of unoccupied blocks under normal traffic conditions [21].

6. Results and Discussion

Experiments ran across three scenarios: normal (typical weekday timetable), moderate disturbance (random primary delays of 0–12 min), and severe disturbance (partial two-block blockage for 30 min). Each scenario was executed for 50 episodes; results report means with 95% confidence intervals. Fig. 4 illustrates section throughput.

6.1. Section Throughput

Table 3 presents the mean number of trains completing the monitored section within the schedule tolerance of ±3 min. The AI controller achieves a 23.1% throughput improvement over the rule-based baseline under normal conditions, rising to 39.4% under severe disturbance. The predictive module's ability to re-sequence trains before a blockage onset proves particularly valuable in the most challenging scenario.

Table 3. Section Throughput—Mean Trains Completing within Schedule Tolerance (Higher is Better)

Scenario	Rule-Based	Human Dispatcher	AI Controller (Proposed)
Normal	41.2 ± 0.6	42.8 ± 0.9	50.7 ± 0.7 (+23.1%)
Moderate Disturbance	33.5 ± 1.1	36.2 ± 1.4	43.1 ± 1.0 (+28.7%)
Severe Disturbance	24.1 ± 1.8	27.4 ± 2.1	33.6 ± 1.6 (+39.4%)

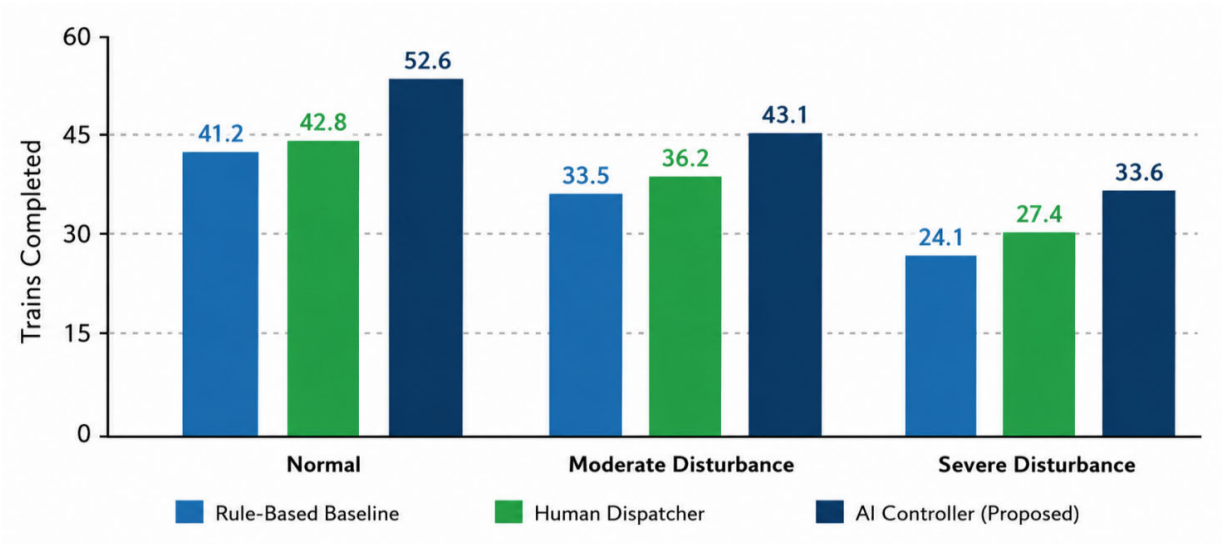

Fig. 4. Section throughput—Rule-based vs. human dispatcher vs. AI controller.

6.2. Delay Amplification

Delay amplification is the ratio of total cumulative exit delay to total injected primary delay. A ratio of 1.0 implies no amplification; values above 1.0 indicate knock-on effects. Table 4 shows that the AI controller reduces delay amplification by 41% relative to the baseline under moderate disturbance. Even under the severe blockage scenario, a 31% reduction is achieved—operationally significant given the inescapable

capacity constraint imposed by the blockage itself.

Table 4. Delay Amplification Ratio (Lower is Better)

Scenario	Rule-Based	Human Emulator	AI Controller
Normal	1.08 ± 0.03	1.05 ± 0.04	1.02 ± 0.02
Moderate Disturbance	1.74 ± 0.12	1.58 ± 0.10	1.31 ± 0.08 (−41%)
Severe Disturbance	2.43 ± 0.21	2.18 ± 0.17	1.68 ± 0.13 (−31%)

6.3. Safety and Key Behavioural Patterns

The safety shield activated in 0.3% of decision steps under normal conditions and 1.1% under severe disturbance, intercepting every unsafe speed target. Zero safety violations were recorded across all episodes. Three recurring AI behaviors were identified: (i) anticipatory speed reductions up to 20 min ahead of predicted congestion to prevent bunching; (ii) priority inversion at passing loops, occasionally granting delayed freight trains precedence over express services to release downstream passenger paths; and (iii) dynamic dwell compression at intermediate stations when a closely following train would otherwise be forced to hold under a signal.

7. Conclusion and Future Scope

This paper presented a comprehensive AI-powered train traffic control framework targeting section throughput maximization on a mixed-service railway corridor. The architecture combines a MARL controller, an LSTM congestion predictor, and a formal safety shield, deployable above legacy interlocking infrastructure without any hardware replacement. Simulation results on a 187 km single-track corridor demonstrated 23–39% throughput improvements and a 41% reduction in delay amplification relative to a rule-based baseline, with zero safety violations recorded throughout all evaluation episodes.

The broader implication is that significant latent capacity exists within most railway networks and that a meaningful share of it can be unlocked through software-based intelligence alone—without waiting for costly infrastructure upgrades. As sensor networks mature and regulatory frameworks for AI in safety-critical systems solidify, AI-powered precise train traffic control is well-positioned to become a mainstream component of railway operations management.

Future directions include: a shadow-mode field trial on a low-traffic regional corridor to validate simulation-to-reality transfer; extension to multi-corridor network-level MARL coordination; integration of passenger demand forecasting into the objective function; a probabilistic safety shield accounting for degraded adhesion under adverse weather; and coupling traffic control with energy management to deliver measurable environmental co-benefits alongside the demonstrated throughput gains.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Omkar Bhasme and Tejas Pasalkar conducted the research; Dipali Bhusari analyzed the data; Paras Bhandari wrote the paper; all authors had approved the final version.

Acknowledgement

The authors extend their sincere gratitude to the faculty members of KJEI's Trinity Academy of Engineering for their guidance throughout this research, to railway operations practitioners who provided valuable insights into real-world dispatching constraints, and to the anonymous reviewers whose constructive

feedback substantially improved the clarity and rigor of this paper.

References

[1] McKinsey & Company. (2022). The rail sector's journey to net zero: A strategic perspective on European railway performance. Retrieved from https://www.mckinsey.com/

[2] Ye, H., Sun, Z., Fang, J., & Jiang, Z. (2019). Predictive train control using integrated deep learning. *IEEE Access*, *7*, 24753–24763. https://doi.org/10.1109/ACCESS.2019.2898973

[3] Dollevoet, T., Huisman, D., Schmidt, M., & Schöbel, A. (2012). Delay management with rerouting of passengers. *Transportation Science*, *46(1)*, 74–89. https://doi.org/10.1287/trsc.1110.0375

[4] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint, arXiv:1707.06347.

[5] Abril, M., Barber, F., Ingolotti, L., Salido, M. A., Tormos, P., & Lova, A. (2008). An assessment of railway capacity. *Transportation Research Part E: Logistics and Transportation Review*, *44(5)*, 774–806. https://doi.org/10.1016/j.tre.2007.04.001

[6] Bešinović, N., Goverde, R. M. P., Quaglietta, E., & Roberti, R. (2022). Neural network-based prediction for train rescheduling. *Transportation Research Part C: Emerging Technologies*, *138*, 103605. https://doi.org/10.1016/j.trc.2022.103605

[7] Carey, M., & Lockwood, D. (1995). A model, algorithms and strategy for train pathing. *Journal of the Operational Research Society*, *46(8)*, 988–1005. https://doi.org/10.1057/jors.1995.135

[8] Chang, C. S., & Sim, S. S. (1997). Optimising train movements through coast control using genetic algorithms. *Proceedings of IEE—Electric Power Applications*, *vol. 144, no. 1*. (pp. 65–73). https://doi.org/10.1049/ip-epa:19970818

[9] Corman, F., D'Ariano, A., Pacciarelli, D., & Pranzo, M. (2010). A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B: Methodological*, *44(1)*, 175–192. https://doi.org/10.1016/j.trb.2009.05.004

[10] European Union Agency for Railways (ERA). (2019). *ERTMS/ETCS System Requirements Specification* (Subset-026 v3.6.0). Retrieved from https://www.era.europa.eu/

[11] Goverde, R. M. P. (2005). Punctuality of railway operations and timetable stability analysis [Doctoral dissertation, Delft University of Technology]. TU Delft Repository. Retrieved from https://repository.tudelft.nl/

[12] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9(8)*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[13] Wang, Y., Tang, T., Ning, L., & van den Boom, T. (2018). Passenger demand oriented train scheduling and rolling stock circulation planning for an urban rail transit line. *Transportation Research Part B: Methodological*, *118*, 193–212. https://doi.org/10.1016/j.trb.2018.10.006

[14] Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems (NeurIPS), 30*, 6379–6390.

[15] Yin, J., Yang, L., Tang, T., Gao, Z., & Ran, B. (2017). Energy-efficient metro train rescheduling. *Transportation Research Part B: Methodological*, *91*, 178–210. https://doi.org/10.1016/j.trb.2016.07.005

[16] International Union of Railways (UIC). (2004). *UIC Code 406: Capacity*. Retrieved from https://uic.org/

[17] Zhang, Q., Han, B., & Li, D. (2008). Modeling and simulation of passenger alighting and boarding movement in Beijing metro stations. *Transportation Research Part C: Emerging Technologies, 16(5)*, 635–649. https://doi.org/10.1016/j.trc.2008.01.002

[18] Lüthi, M., Medeossi, G., & Nash, A. (2009). Increasing railway capacity and reliability through integrated real-time rescheduling. *Transportation Research Record, 2117*, 56–65. https://doi.org/10.3141/2117-08

[19] Pachl, J. (2002). Railway Operation and Control. VTD Rail Publishing.

[20] Zhong, Q., Zhao, Y., Zhu, X., & Liu, H. (2021). Attention-based reinforcement learning for train regulation in metro systems. *IEEE Transactions on Intelligent Transportation Systems, 23(7)*, 8279–8292. https://doi.org/10.1109/TITS.2021.3055525

[21] Quaglietta, E., Pellegrini, P., Goverde, R. M. P., Albrecht, T., Jaekel, B., & Hansen, I. A. (2016). Impact of a stochastic and dynamic setting on railway dispatching stability. *Journal of Rail Transport Planning & Management, 6(1)*, 1–22. https://doi.org/10.1016/j.jrtpm.2015.11.002