# AEROGRAM: Adaptive Environment & Rerouting Optimiser with GMM-augmented LSTM Airspace Model


Cristian - Augustin Susanu
*Computer Science Dept.*
*Vrije Universiteit Amsterdam*
Amsterdam, Netherlands
c.susanu@student.vu.nl

Claudia Raibulet
*DISCo - Dip. di Informatica,*
*Sistemistica e Comunicazione,*
*University of Milano-Bicocca*, Milan, Italy
claudia.raibulet@unimib.it

Ilias Gerostathopoulos
*Computer Science Dept.*
*Vrije Universiteit Amsterdam*
Amsterdam, Netherlands
i.g.gerostathopoulos@vu.nl



***Abstract*—The accelerating growth of global air traffic is widening the gap between traditional air-traffic management (ATM) tools and the real-time, data-intensive decisions modern control towers must make. We introduce AEROGRAM[1,2], an open-source artefact that merges a combination of Long Short-Term Memory (LSTM) and Gaussian Mixture Model (GMM) capacity predictor with a dashboard-driven MAPE-K adaption loop. Developed and calibrated for Amsterdam Schiphol Airport, AEROGRAM continuously ingests live Automatic Dependent Surveillance–Broadcast (ADS-B), Advanced Surface Movement Guidance and Control System (A-SMGCS) and METAR feeds, evaluates three interchangeable strategies (rule-based baseline, pattern-based GMM, deep LSTM) and surfaces rerouting advice, delay forecasts and uncertainty thresholds in an interactive Graphical User Interface (GUI). Experimental results on Schiphol traffic scenarios show that the LSTM based adaptive strategy cuts average delay by 33% and sustains 85–90% efficiency during peak hours, while the GMM alternative delivers moderate gains with half the compute footprint and the baseline remains lightweight but least effective. By packaging monitoring, analysis, planning, execution and knowledge into a transparent, modular simulator, AEROGRAM lowers the barrier to reproducible ATM research and provides air-traffic controllers with an interpretable decision-support tool.**



***Index Terms*—AEROGRAM, Air Traffic Management, self-adaptive systems, trajectory optimization, machine learning, LSTM, GMM, aviation safety, Schiphol Airport.**


## I. Introduction

The post-pandemic recovery of global air travel is stretching today's ATM systems. Controllers now coordinate denser flight paths, volatile weather and tighter environmental limits while safeguarding operations in real time. Passenger traffic surpassed 4.2 billion and departures reached 35 million in 2023, yet the accident rate remained at 1.87 per million departures [1]. Nonetheless, loss-of-control-in-flight (LOC-I) is still the leading fatal-risk category, showing that current safety margins are fragile [1]. Regulators are reacting: EASA's AI Roadmap 2.0 positions machine learning (ML) at the centre of 4-D trajectory prediction, hybrid conflict detection and human-centric decision support [2], underscoring a widening gap between rule-based ATM software and the demands of future airspace.

Systematic reviews [3] point to data-driven models as a remedy: LSTMs dominate short-term trajectory forecasting [4], while GMMs capture multimodal traffic regimes [5]. Yet fragmented datasets, substantial computational costs and limited model interpretability still constrain operational integration. The challenge is therefore not only to design better algorithms, but also to present them in a form controllers can trust under rigorous real-time constraints.

Few frameworks tackle trajectory prediction and optimisation for safety-critical ATM. **AEROGRAM** is a unified, standalone platform that brings the entire adaptive-rerouting loop into one dashboard. It continuously ingests live ADS-B tracks, METAR weather, surface-movement data and operator stress tests, then runs a LSTM/GMM engine inside a closed MAPE-K loop to detect conflicts, forecast runway throughput and quantify uncertainty. The GUI illustrates runway status and weather, synchronised capacity charts, clustering views and control panels for replaying traffic, tweaking weather or toggling between baseline and ML-enhanced strategies. Detailed logs archive every prediction, decision and rerouting action for later analysis.

Three design features make **AEROGRAM** suitable for both research and operational trials:

1) **Adaptive simulation dashboard**: injects live disruptions as weather cells, runway outages or demand surges, while recording the system's responses for replay.
2) **Modular strategy catalogue**: allows baseline, LSTM or GMM strategies to be swapped at run time, enabling systematic comparison under identical conditions.
3) **Airport adapters**: connectors that map local ADS-B, A-SMGCS and METAR feeds to AEROGRAM's schema, supporting prompt deployment at almost any airport.

AEROGRAM complements existing platforms, such as SWITCH [6], which demonstrates real-time ML model switching with a Kibana dashboard but omits aviation-specific metrics such as predicted delay and conflict-lead times. On

[1] AEROGRAM is available as open-source software here.
[2] A Demo video can be found here.

the other hand, Wildfire-UAVSim [7] excels in drone-based wildfire scenarios but operates on vastly different spatial and temporal scales, uniquely addressing the split-second, high-density dynamics of terminal airspace. Last but not least, DARTSim [8] simulates Unmanned Aerial Vehicle (UAV) reconnaissance missions under partial observability, modelling sensing errors, tactic latency and survival-versus-performance trade-offs, but focuses on single-mission scenarios and lacks support for continuous, high-density traffic flows, real-time rerouting and the aviation-specific interpretability metrics. By combining a low-latency hybrid model with a cross-airport compatible simulator, AEROGRAM lowers the barrier to systematic studies of ML-driven trajectory optimisation. We anticipate that this artefact will accelerate progress from promising prototypes to practical decision-support tools in the control tower, helping to keep future skies efficient and safe.

In conclusion, AEROGRAM mitigates the gap between rule-based ATM and forthcoming high-density operational airspace by uniting LSTM- and GMM-driven models in a real-time, dashboard-based MAPE-K loop that incorporates live ADS-B, METAR and surface-movement data to deliver transparent, interactive trajectory forecasts and rerouting advice. The prototype presented in this paper is explicitly developed and calibrated for Amsterdam Schiphol Airport (IATA: AMS, ICAO: EHAM) [9]. Schiphol's six-runway configuration, high-density traffic patterns and well-documented open-data feeds make it an ideal test-bed for evaluating adaptive ATM strategies; all dashboards, capacity tables and experiments reported herein therefore reflect Schiphol-specific operating rules and traffic statistics.

The rest of this paper is structured as follows: Section II reviews related work; Section III describes the architecture and implementation; Section IV presents our experiment workflow and evaluation scenarios; Section V discusses the results, trade-offs and limitations; and Section VI concludes with our key contributions and future directions.

## II. Related Work

Several research exemplars have introduced self-adaptive and ML-enabled systems in dynamic, safety-critical domains, but each addresses only part of the challenges AEROGRAM aims to integrate.

***SWITCH*** [6] provides a cloud-native web service for dynamic ML model switching to maintain QoS in generic ML-Enabled Systems. It features real-time data handling, comprehensive logging and an interactive Kibana dashboard that helps researchers evaluate switching strategies in object-detection scenarios. However, it treats adaptation at the service level and omits domain-specific metrics such as predicted delay or conflict-lead time, which are crucial for air-traffic applications.

***Wildfire-UAVSim*** [7] formalizes the problem of tracking a spreading wildfire under partial observability and offers a highly configurable simulator for evaluating UAV-based adaptation strategies. It lets users vary forest density, fire and smoke propagation and UAV behaviors, but its spatial and temporal scales, designed for low-density, long-duration forest monitoring differ substantially from the split-second, high-density dynamics of terminal airspace.

***DARTSim*** [8] targets smart cyber-physical systems by simulating a UAV reconnaissance mission in a hostile environment, emphasizing self-adaptation challenges like sensing uncertainty, tactic latency and incomparable safety versus performance objectives. It offers both a TCP-based and library interface for seamless integration of external adaptation managers and supports rapid, deterministic simulation runs. Yet its focus remains on high-level, single-mission trade-offs rather than continuous, multi-flight traffic flows.

Together, these tools illustrate key principles such as dynamic model management, partially observable environment simulation and real-time self-adaptation, but none provides an integrated, air-traffic-specific platform with interpretable ML pipelines, live rerouting advice and flexible airport data adapters in the manner AEROGRAM does.

## III. Architecture and Implementation

AEROGRAM's architecture is organized into three tightly integrated subsystems (see Figure 1):

- ***Graphical User Interface***, representing the medium through which the user interacts with the system.
- ***Managing System***, implementing a MAPE-K loop with the ML and trajectory processing engine.
- ***Managed System***, containing the relevant airport, flight and weather data, offering a flight scheduling service.

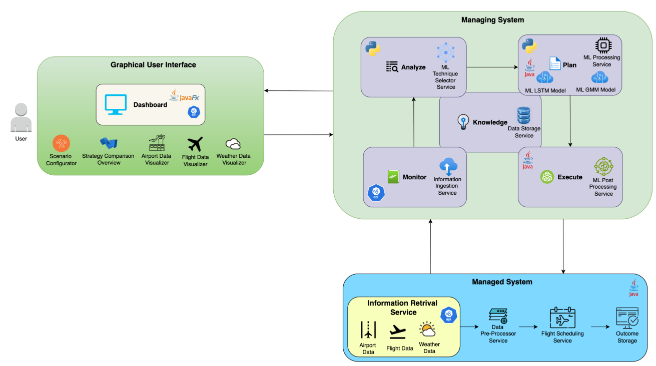


Fig. 1. System Architecture

### A. *Graphical User Interface (GUI)*

The GUI, (see Fig. 2) is implemented through a JavaFX-based [10] framework, providing a user-centric interface that displays all key AEROGRAM metrics, strategy controls, and real-time data feeds in an integrated view.

*1)* ***Strategy Comparison Overview****:* allows the user to see AEROGRAM's real-time behaviour under baseline rules, the LSTM forecaster and the GMM clusterer. Since the artefact focuses on adaptive trajectory optimisation and safety improvement, it shows hourly counts of accepted vs cancelled flights, delay and efficiency metrics. The baseline strategy, a purely rule-based controller that follows fixed ICAO runway-capacity tables (Table II), scaled for weather and night factors, serves as a non-adaptive reference against which the adaptive capabilities of the LSTM and GMM can be measured. The two ML methods were chosen for their complementary strengths: the LSTM captures sequential traffic dependencies and provides precise throughput forecasts even in highly dynamic situations, while the GMM provides probabilistic clustering of traffic regimes with explicit uncertainty bounds for robust decisions in ambiguous or mixed-mode situations [3].

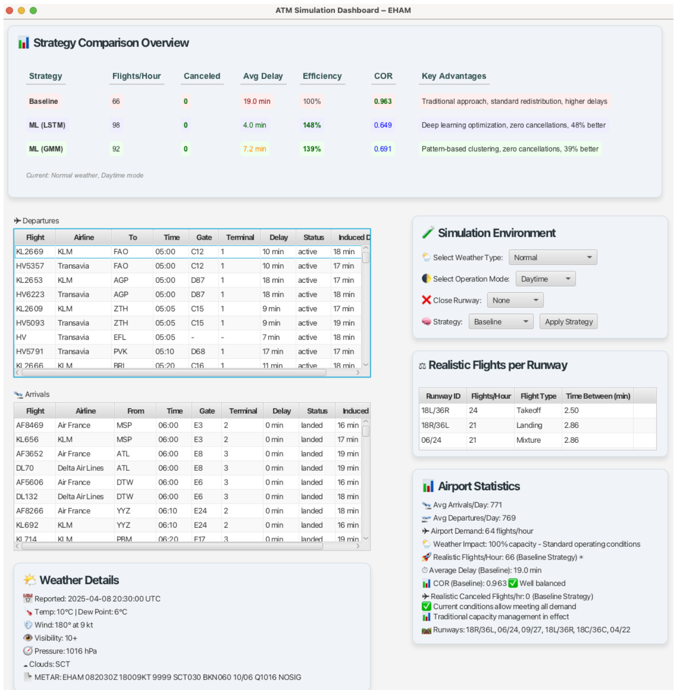


Fig. 2. Graphical User Interface

*2) **Departure and Arrivals tables**:* present typical flight-related information within the given day obtained through the Aviation Stack API [11]. The last column of these 2 tables, the Induced Delay metric represents the improvements brought through each strategy. For ease of result reproductibility, the response body from this API has been saved in 2 separate JSON files, one for departures and one for arrivals.

*3) **Simulation Environment**:* represents the operational side of the GUI, the place where the user can experiment with various operating conditions. Table I presents the scaling factors applied for the weather and operation modes, as well as in case a runway is closed down. These scaling rules apply to all strategies and their role is to generate a realistic environment for the user, reproducing real-case scenarios.

*4) **Realistic Flights per Runway**:* The runway-related information for the airport of interest, Schiphol for this paper's case is extracted from The Aviation Weather Centre's API [12]. Table II further details the available runways, their capacity in terms of flights/hour and the flight types they can accommodate, which are presented in accordance with [9], being adapted for the artefact's simulation environment needs.

*5) **Airport Statistics**:* showcase a large variety of information for the chosen strategy (see Fig. 2). There is a mix

TABLE I
SIMULATION-ENVIRONMENT CONTROLS AND SCALING RULES

| Parameter | Option | Rule applied in Baseline / LSTM / GMM |
|---|---|---|
| Weather | Excellent<br>Normal<br>Bad<br>Bad + Precipitation<br>Bad + Strong wind | Capacity $\times 1.10$, Delay $\times 0.70$<br>Capacity $\times 1.00$, Delay $\times 1.00$<br>Capacity $\times 0.85$, Delay $\times 1.50$<br>Capacity $\times 0.80$, Delay $\times 1.80$<br>Capacity $\times 0.70$, Delay $\times 2.00$ |
| Operation | Daytime<br>Nighttime | Demand $\times 1.00$ (all models)<br>Baseline: cap. $\times 0.75$<br>LSTM: night$=1$ ($\approx 12\%$ lower cap.)<br>GMM: night cluster, cap. $\times 0.80$ |
| Runway closure | None<br>Any runway ID | No capacity loss (all models)<br>30 % lost + 70 % redistributed (Baseline);<br>0 % lost + 100 % redistributed (LSTM);<br>pattern-based redistribution (GMM) |
| Strategy | Baseline<br>ML (LSTM)<br>ML (GMM) | Pure rule set (no ML)<br>Deep LSTM throughput predictor activated<br>Gaussian-mixture pattern optimiser activated |

TABLE II
SCHIPHOL RUNWAYS CONFIGURATION [9]

| Runway Identifier | Capacity (flights/hr) | Flight Type |
|---|---|---|
| 18R_36L | 36 | Landing |
| 06_24 | 36 | Mixture |
| 18C_36C | 34 | Mixture |
| 18L_36R | 40 | Takeoff |
| 09_27 | 25 | Mixture |
| 04_22 | 10 | Takeoff |

between static information (daily average arrivals/departures, available runways) and dynamic results for each strategy. Airport demand is first taken as the 24-hour average of arrivals plus departures and then scaled for weather severity and reduced night-time operations, as detailed in Table I. For the baseline strategy, the per-hour demand is multiplied by the weather and night-time factors, rounded and capped at the total available runway capacity, accounting for any closures, with any unmet demands recorded as cancellations. The two ML approaches instead reconfigure runways to fully restore capacity, so their hourly throughput equals the sum of their runway capacities and cancellations are zero. Average delays for all strategies are calculated by a unified penalty model that factors in weather, operation mode, and runway availability.

*6) **Weather Details**:* present comprehensive weather data, obtained from the Aviation Weather Center [12] for the chosen simulation, offering a METAR reading as well. For simplicity of reproducing the artefact, the weather information has been saved in separate JSON files to ensure all the weather conditions can be emulated successfully within the simulation.

## B. *Managing System*

The Managing System is a modular Java micro-service built around the MAPE-K (Monitor, Analyze, Plan, Execute, Knowledge) loop [13] and represents the adaptive module between the GUI and the managed system layers. **Monitor** continuously ingests live ADS-B tracks, surface-movement events, METAR updates and user inputs. During **Analyze**, the service selects one of the 3 strategy paths. If LSTM is chosen, LSTMStrategyService builds a 24-feature vector containing weather flags, night indicator, runway status, rolling demand, airline tags and sends it via gRPC to a Python service hosting a three-layer, attention-augmented LSTM ($R^2 \approx 0.84$) and the model returns a per-runway capacity vector. Alternatively for the GMM strategy, GMMStrategyService forwards an hour-level traffic snapshot to a 4 cluster GMM; the chosen centroid provides the capacities, while posterior weights yield an uncertainty score that the planner down-weights when confidence falls below 0.6. For the baseline strategy, BaselineStrategyService reverts to ICAO capacity (Table II), scaled by weather and night-time factors. **Plan** merges the selected capacities with the live arrival queue to generate spacing, resequencing or holding instructions and **Execute** dispatches these actions to both the GUI and the flight-scheduling service. Finally, **Knowledge** archives every feature vector, prediction, decision and outcome for later retraining, performance comparison and bidirectional feedback. In this way the Managing System transforms raw operational data into actionable guidance while exposing clear, real-time insights to the user.

### C. *Managed System*

The Managed System focuses on core data handling and deterministic flight-scheduling. An Information Retrieval Service continuously polls external APIs for runway status, ADS-B tracks, flight schedules and METAR weather, then hands the raw feeds to a Data Pre-Processor Service that time-stamps, de-duplicates and merges them into a unified event stream. This stream is passed to a Java-based Flight Scheduling Service, which applies the airport's published separation rules and slot priorities to generate the next-hour arrival/departure sequence. The resulting plan, along with key counters (queue length, average spacing) is persisted in the Outcome Storage and pushed upward to the Managing System, where the ML-powered Analyze phase can compare its own capacity forecasts against this baseline schedule. By keeping data ingestion and scheduling in a clean, self-contained layer, airports can swap in new feeds or tweak slot allocation rules without touching the adaptation logic that lives higher in the stack.

In summary, our modular, API-based architecture, separating the Java dashboard and strategy services, MAPE-K adaptation logic, and Python ML training scripts and inference engines enables rapid experimentation, reproducible comparisons and easy deployment at any airport with ADS-B, A-SMGCS, and METAR feeds.

## IV. Experiment Workflow

To evaluate AEROGRAM under realistic operational conditions, we define 2 representative adaptation scenarios, presented in Table III. Scenario $S_1$ exposes the airport to intense, fast-moving weather cells, prompting the system to redistribute arrivals toward unaffected runways. Scenario $S_2$ simulates an abrupt runway outage, requiring rapid traffic reallocation across the remaining configuration. Both scenarios are formulated to probe the artefact's ability to ensure 3 core objectives: safety, delay mitigation and capacity efficiency, while mirroring the decision pressure of a live control tower.

TABLE III

Adaption scenarios supported by AEROGRAM

| Scenario | Type of uncertainty | Type of adaptation | Type of requirement |
|---|---|---|---|
| $S_1$ | *Unpredictable environment:* **severe weather conditions** | Capacity scaling, distributing flights to runways not affected by the particular weather cells. | Safety, delay enhancement, capacity efficiency |
| $S_2$ | *Operational capacity uncertainty:* **runway closure** | Runway redistribution: flights are re-assigned and new available runways are opened. | Safety, delay enhancement, capacity efficiency |

Performance is quantified through a concise set of domain-specific metrics. Safety is tracked via the Capacity-Oversubscription Ratio (COR) [14], which identifies overloads by dividing queued flights by the strategy's own capacity estimate (Eq. 1). Delay enhancement combines mean arrival delay with the additional delay induced by each tested strategy and capacity efficiency is measured by the number of realistically handled flights per hour together with the count of cancellations. These metrics, summarised in Table IV, supply a balanced evaluation matrix that links tactical decisions to passenger-relevant outcomes.

TABLE IV

Quality attributes and metrics

| Quality attribute | Metric |
|---|---|
| Safety | Capacity-oversubscription ratio (COR) (Eq. 1) |
| Delay enhancement | Mean arrival delay (s); Induced delay (s) |
| Capacity efficiency | Number of realistic flights per hour compared to the airport demand metric; Number of canceled flights per hour. |

$$\text{COR} = \frac{\text{Demand}}{\text{Predicted Capacity}} \tag{1}$$

All scenarios run inside AEROGRAM's interactive dashboard, which merges live ADS-B, METAR and surface-movement feeds with a controllable simulation layer. This setup lets researchers inject weather shifts, alter runway availability or switch between the rule-based baseline, pattern-based GMM and deep LSTM strategies on demand, while recording every forecast, decision and outcome for replay and analysis.

## V. Discussion

In this section, we analyze the contrasting performance of 3 ATM strategies implemented in AEROGRAM: the rule-based baseline, the pattern-based GMM and the predictive LSTM. We first examine the high-level summary in Fig. 3, then delve into the detailed metrics in Fig. 4.

Fig. 3 presents 4 key panels. In **Average Flight Delay by Strategy**, the baseline yields about 20 min of delay per flight, GMM reduces this by 20% to 16 min and LSTM achieves a 33% reduction to 13.3 min. The **Weather Resilience** chart shows that while all strategies maintain 100% efficiency under Excellent and Normal conditions, efficiency drops in adverse weather to 70/60/50 % for baseline, 80/75/70 % for GMM and a robust 90/85/80 % for LSTM. In **Adaptation Speed**, log-scale response times to disturbances range from 15–25 min for baseline, 5–15 min for GMM and sub-2 min for LSTM,

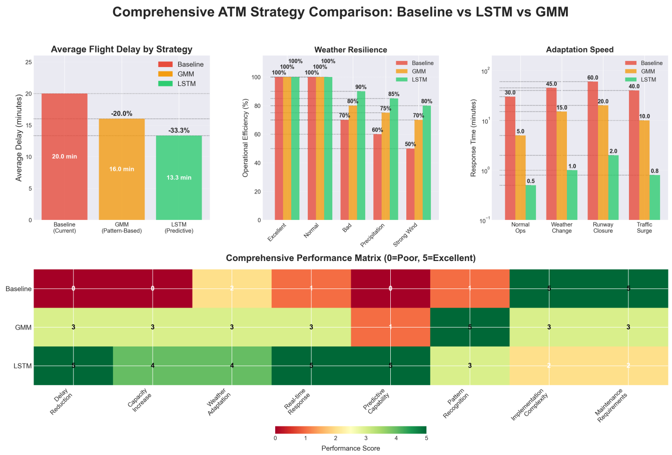


Fig. 3. Comprehensive ATM strategy comparison

highlighting LSTM's agility. Finally, the **Comprehensive Performance Matrix** captures 8 dimensions: delay reduction, capacity increase, weather adaptation, real-time response, predictive capability, pattern recognition, implementation complexity and maintenance requirements, showing LSTM's 4–5 ratings on performance metrics but lower scores on complexity and maintenance, GMM's consistent mid-range performance and the Baseline's simplicity strengths.

Building on this overview, Fig. 4 offers 10 detailed views. The **Prediction Performance** panel measures $R^2$, MAE, RMSE and accuracy, where LSTM leads ($R^2 = 0.89$, MAE≈2.8 min, RMSE≈4.2 min, 92%), GMM sits in the middle ($R^2 \approx 0.70$, MAE≈5.2 min, RMSE≈8.1 min, 78%) and baseline falls behind ($R^2 = 0.45$, MAE≈8.5 min, RMSE≈12 min, 65%). **Resource Requirements** shows LSTM's heavier footprint (60% CPU, 4 GB memory, 1 GB storage, 1 Mbps) versus GMM (40%/2 GB/0.5 GB/0.5 Mbps) and baseline (20%/0.5 GB/0.1 GB/0.1 Mbps). The **Key Performance Indicators** chart aggregates prediction accuracy, delay reduction, real-time adaptability and weather resilience into normalized scores, again favoring LSTM. The **Error Distribution** histogram confirms tighter error concentration for LSTM. In the middle row, **Weekly Performance Stability** tracks delay variance (10–15 min for LSTM, 15–25 min for GMM, spikes to 35 min for baseline) and **24-Hour Adaptive Performance** plots diurnal efficiency, with LSTM sustaining 85–90% even at peaks. The **Computational Speed** plot (log-scale) reveals very different scaling behaviors: the baseline strategy's processing time grows from roughly 50 ms for a single flight to about 10,000 ms for a full-day batch; GMM sits in the middle, climbing from 10 ms up to 3,000 ms and LSTM remains by far the fastest, starting under 1 ms on a single flight and only reaching 30 ms even for the 2,000-flight batch. The The **Model Complexity** bubble chart plots each strategy's parameter count on a logarithmic y-axis: baseline at roughly $1 \times 10^3$ parameters, GMM at about $5 \times 10^3$ and LSTM at approximately $5 \times 10^4$, while bubble area encodes relative training time (small for baseline, medium for GMM, large for LSTM), clearly illustrating LSTM's substantially greater model capacity and training cost compared to the lighter-weight alternatives. The bottom row highlights long-term learning and scalability. In **Long-term Performance Gains**, baseline stays at 0%, GMM rises to $\sim$ 25% by two years then plateaus and LSTM steadily climbs to $\sim$ 37% by year 5. In **Performance Scaling with Data Size**, baseline remains $\sim$ 65% accuracy, GMM tops out near 80% by $10^4$ flights and LSTM accelerates past 85% around $10^5$ and reaches $\sim$ 92% at $10^6$. These charts underline LSTM's continued improvement over time and data, GMM's earlier plateau and baseline's fixed ceiling.

In conclusion, these figures reveal clear trade-offs: the baseline strategy offers minimal computational overhead and implementation simplicity at the cost of limited performance, GMM strikes a middle ground with moderate gains and resource requirements and LSTM delivers superior predictive accuracy, resilience and responsiveness but requires greater computing resources and maintenance effort. The AEROGRAM dashboard thus provides a transparent decision-support tool, enabling air traffic controllers to select the strategy best aligned with their operational constraints and long-term objectives.

## VI. Conclusion

This paper introduced AEROGRAM, a self-adaptive ATM artefact that unifies procedural, pattern-based and predictive strategies within a real-time MAPE-K loop. By combining live ADS-B, METAR and surface-movement feeds with a dashboard-centric GUI, the system delivers transparent capacity forecasts, rerouting advice and rich performance analytics. Our evaluation shows that the LSTM strategy achieves the greatest benefits, cutting average delay by 33 % and sustaining 85–90 % efficiency during rush hours, while the GMM strategy offers a balanced alternative with moderate resource needs. The baseline remains computationally lightweight but lags in every performance dimension. Together, the results validate AEROGRAM's design goals: a customizable simulator for reproducible experimentation, a clear decision-support interface for air-traffic controllers and an extensible architecture that accommodates multiple ML pipelines without sacrificing interpretability.

**Future work** can expand AEROGRAM in three directions. First, we will introduce a registry of airport adapters so that local runway layouts, separation rules and data feeds can be mapped to a common schema and users can switch seamlessly between airports from within the GUI. Second, a policy layer will be added to recommend the most cost-effective strategy between baseline, GMM, LSTM or a hybrid on the fly, balancing accuracy against computational cost for each airport-weather-demand context. Finally, containerising the Managing System and moving the heavier models to GPU micro-services in the cloud will enable sub-second inference even when multiple airports are running concurrently. By pursuing these directions, AEROGRAM can evolve from a

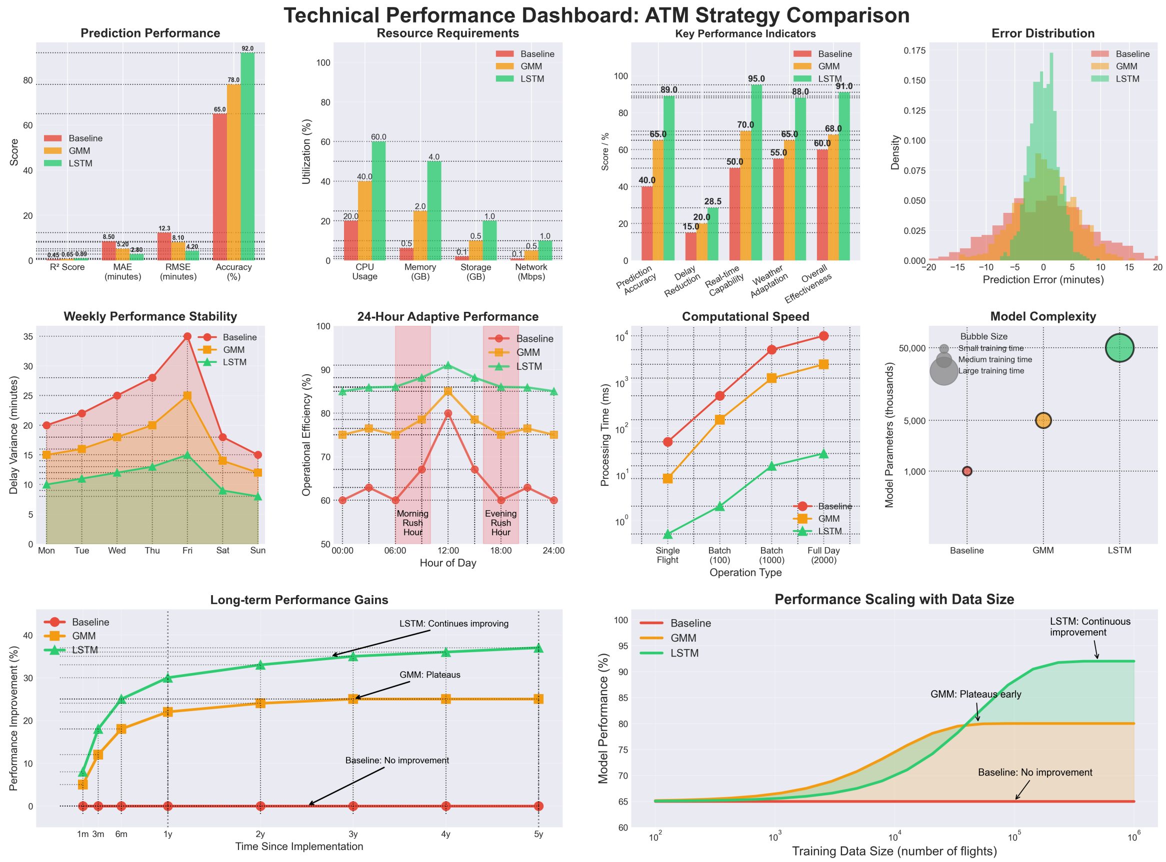


Fig. 4. Technical Performance Dashboard comparing Baseline, GMM, and LSTM strategies across various metrics.

single-airport research demonstrator into a versatile, cloud-ready platform that offers adaptive guidance for any airport a controller selects, thereby advancing both the state of ATM research and its operational readiness.

## References


[1] International Civil Aviation Organization, "ICAO Safety Report: 2024 Edition," International Civil Aviation Organization (ICAO), Montréal, Canada, 2024, https://www.icao.int/safety/Documents/ICAO_SR_2024.pdf

[2] European Union Aviation Safety Agency, "Artificial Intelligence Roadmap 2.0: Human-Centric Approach to AI in Aviation," EASA, May 2023, https://www.easa.europa.eu/ai

[3] C.A. Susanu, C. Raibulet, S. Alam, G. Lulli, "Machine Learning in Air Traffic Management for Trajectory Optimization and Aviation Safety: A Systematic Literature Review," SSRN, May 2025, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5256104

[4] K. Cao, T. Zhang, and J. Huang, "Advanced hybrid LSTM-transformer architecture for real-time multi-task prediction in engineering systems," Scientific Reports, vol.14, no.4890, 2024, https://doi.org/10.1038/s41598-024-55483-x

[5] Y. El-Laham, N. Dalmasso, E. Fons, and S. Vyetrenko, "Deep Gaussian Mixture Ensembles," in Proceedings of the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023), Proc. Machine Learning Research, vol. 216, pp. 549–559, 2023, https://proceedings.mlr.press/v216/el-laham23a/el-laham23a.pdf

[6] A. Marda, S. Kulkarni, and K. Vaidhyanathan, "SWITCH: An Exemplar for Evaluating Self-Adaptive ML-Enabled Systems," in *Proceedings of the 19th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2024)*, ACM, pp. 143–149, 2024, https://doi.org/10.1145/3643915.3644105

[7] E. Vilchez, J. Troya, and J. Cámara, "Wildfire-UAVSim: An Exemplar for Evaluation of Adaptive Cyber-Physical Systems in Partially-Observable Environments," in Proceedings of the 19th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2024), ACM, pp. 63–69, 2024, https://doi.org/10.1145/3643915.3644109

[8] G. A. Moreno, C. Kinneer, A. Pandey, and D. Garlan, "DARTSim: An Exemplar for Evaluation and Comparison of Self-Adaptation Approaches for Smart Cyber-Physical Systems," in Proceedings of the IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), pp. 133–139, 2019, https://squareslab.github.io/materials/MorenoDART2019.pdf

[9] Royal Schiphol Group, "Flight paths and runway use," Amsterdam Airport Schiphol, https://www.schiphol.nl/en/schiphol-as-a-neighbour/flight-paths-and-runway-use/

[10] OpenJFX Community, "JavaFX: Open Source, Next Generation Client Application Platform for Java," OpenJFX, 2024. https://openjfx.io

[11] Aviationstack, "Real-Time Flight Status & Global Aviation Data API," Aviationstack API Documentation, 2024. https://aviationstack.com/documentation

[12] NOAA Aviation Weather Center, "Aviation Weather Data Services API," 2024. https://aviationweather.gov/data/api/

[13] J. Oh, C. Raibulet, and J. Leest, "Analysis of the MAPE-K Loop in Self-adaptive Systems for Cloud, IoT and CPS," in *Proceedings of the ICSOC 2022 Workshops, LNCS 13821, pp. 130–141, 2023, https://doi.org/10.1007/978-3-031-26507-5_11

[14] Z. Chen, H. Xiang, B. Han, Y. Shen, T. Zhou, and F. Zhang, "Research on Runway Capacity Evaluation of General Aviation Airport Based on Runway Expansion System," *Symmetry*, vol. 16, no. 11, art. 1555, 2024, https://doi.org/10.3390/sym16111555