Enhancing Air Traffic Management: A Latent Representation Learning Approach

Saman Mostafavi
Metis Technology Solutions

Charles Ison
Oregon State University

Farzan Masrour Shalmani
Crown Consulting, Inc.

Krishna Kalyanam
NASA

***Abstract*—Efficient management of the National Airspace System (NAS) relies heavily on strategic Traffic Management Initiatives (TMIs), such as Ground Delay Programs (GDPs) and Ground Stops (GSs), to balance fluctuating demand with constrained airport capacity during adverse conditions. This paper introduces a representation learning approach designed to assist air traffic managers in identifying and retrieving similar historical TMI events. Using autoencoder-based latent feature extraction, our method jointly optimizes predictive accuracy of TMI triggers while providing meaningful, interpretable representations clustered in latent embeddings. Evaluations conducted using data from three major airports in the New York Metroplex—JFK, EWR, and LGA—demonstrate improved scenario clustering compared to traditional clustering and dimensionality reduction techniques. Our methodology facilitates scenario-driven recommendations, enhancing decision-making precision and interpretability in TMI implementation.**

***Index Terms*—Representation Learning, Traffic Management Initiatives**

I. Introduction

The aviation industry is undergoing significant transformations driven by technological advancements, increasing air traffic demand, and operational complexities. According to forecasts from the Federal Aviation Administration (FAA) [3], general aviation hours flown in the United States are projected to grow by approximately 0.7% annually over the next two decades, with a total growth of 17.4% by 2044. This trend shows variations across aircraft types, with jet aircraft hours increasing at an average annual rate of 2.5%, while fixed-wing piston hours are forecast to decrease by 0.8% annually. This evolving demand landscape emphasizes the critical importance of efficient operational management strategies that maintain safety and effectiveness within the National Airspace System (NAS). In response, aviation modernization initiatives such as the FAA's Next Generation Air Transportation System (NextGen) and Europe's Single European Sky ATM Research (SESAR) have emerged, aiming to enhance airspace capacity, operational efficiency, and safety.

Central to maintaining efficient operations in the NAS are Traffic Management Initiatives (TMIs), strategic actions coordinated by the FAA's Air Traffic Control System Command Center (ATCSCC). TMIs help manage demand and capacity, particularly during adverse weather conditions or unexpected disruptions. Two primary forms of TMIs are Ground Delay Programs (GDPs) and Ground Stops (GSs). GDPs proactively delay flights at their origin airports to manage anticipated capacity constraints at destination airports, thus replacing airborne delays with safer and more manageable ground-based delays. In contrast, GSs implement immediate, temporary restrictions on aircraft departures or arrivals, typically employed under urgent circumstances. Although TMIs effectively mitigate disruptions, selecting their optimal parameters—such as timing, duration, and geographical scope—remains challenging. Traditional decision-making methods rely heavily on expert judgment derived from historical experiences, weather forecasts, current traffic demands, and known constraints. However, reliance on manual processes can lead to inefficiencies, such as unnecessary prolonged delays or premature termination of programs, both increasing operational disruptions and the workload of air traffic controllers [5].

To address these challenges, recent research has leveraged advanced data-driven methodologies, particularly machine learning (ML) techniques. ML has demonstrated success in aviation anomaly detection, integrating trajectory data, environmental factors, and system-level metrics to identify deviations from standard operations [1], [11]. Typical anomaly detection frameworks employ unsupervised learning approaches, such as autoencoders or clustering algorithms, to uncover and interpret operational anomalies [2], and to cluster and detect events in aircraft trajectories [12]. However, these frameworks often focus exclusively on detecting anomalies without facilitating the retrieval of historically similar scenarios to support proactive operational decisions. Similar challenges in recommendation systems have been addressed in other domains [8], using deep latent representations of multimedia content while capturing implicit user-item relationships.

Motivation: Consider the scenario of an air traffic controller faced with rapidly evolving weather events and operational complexities. As part of a decision-making panel, the manager must quickly analyze multiple streams of information, including weather trajectories, traffic demand forecasts, and current operational conditions, to determine appropriate TMI actions. Currently, air traffic managers rely primarily on personal experience and manual analysis of past events to inform their decisions—a process that can be time-consuming and subject to inconsistencies due to human limitations and interpretive differences [5]. The availability of a decision-support tool capable of querying current conditions and retrieving historically similar scenarios—alongside detailed analyses of how previous interventions unfolded—would substantially streamline the decision-making process. Such a tool would not only improve the accuracy and efficiency of operational decisions but also serve as a valuable training resource, enabling managers to examine past decision-making outcomes and systematically refine their strategic judgment.

Contribution: This research proposes a representation learning framework specifically designed to retrieve historical TMI events similar to current operational scenarios. Our method employs an autoencoder-based [7] latent representation learning approach jointly optimized for predictive accuracy and interpretable representations clustered in latent embeddings. Using comprehensive data from the major New York Metroplex airports—John F. Kennedy International Airport (JFK), Newark Liberty International Airport (EWR), and LaGuardia Airport (LGA)—we demonstrate significant improvements in scenario clustering and retrieval compared to traditional dimensionality reduction techniques, including Principal Component Analysis (PCA) and Uniform Manifold Approximation and Projection (UMAP) [10]. By integrating representation learning directly into the TMI decision-making processes, our methodology enhances decision precision, clarity, and operational insight. This approach empowers air traffic managers to make more informed, timely, and effective decisions, ultimately improving the resilience and operational efficiency of the NAS.

II. Methodology

A. Data Preprocessing

Employing the same data gathering and preprocessing techniques described in our previous work [9], historical air traffic data and weather records from three major New York Metroplex airports (JFK, EWR, LGA) during 2017-2025 was collected for representation learning. The sources for this data includes:

- **Terminal Aerodrome Forecast (TAF)**: Four synoptic weather forecasts per day, parsed into hourly fields including visibility, ceiling, wind direction, and speed.
- **Traffic Management Initiative (TMI) Records**: Comprehensive records of Ground Delay Programs and Ground Stops with timestamps, and primary cause classifications.
- **Aviation System Performance Metrics (ASPM)**: Hourly operational metrics including arrival and departure counts, taxi times, and delay statistics.
- **Notices to Airmen (NOTAMs)**: Structured information on runway closures and airport-specific restrictions, encoded as binary indicator variables.
- **Airspace Flow Programs (AFP)**: Hourly metrics quantifying airborne holding associated with TMIs.
- **Flight Cancellation Logs**

Specific features were then selected from each source using feedback from ATCs on which data most influences TMI decisions. In total, 14 input features were selected and include arrival rates, a eastward wind component, a northward wind component, ceiling, visibility, weather condition intensity, mist, fog, rain, showers, snow, thunderstorms, and vicinity conditions. We denote each feature respectively as: (arr, $wind\_east$, $wind\_north$, $ceiling$, $visibility$, $-$, $+$, BR, ,FG, RA, SH, SN, TS, VC). Each feature is also cleaned to remove outliers, handle missing values, and normalized to zero mean and unit variance to ensure balanced contribution to the model. The wind components are also are computed using wind speed V_s and wind direction V_d:

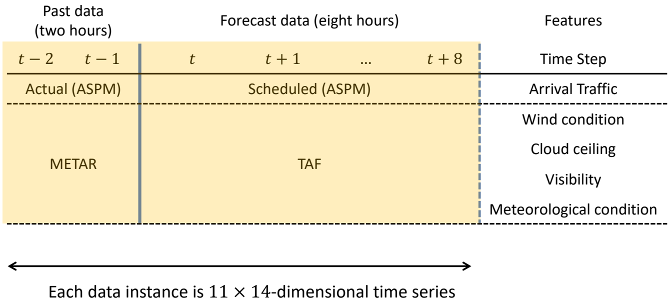


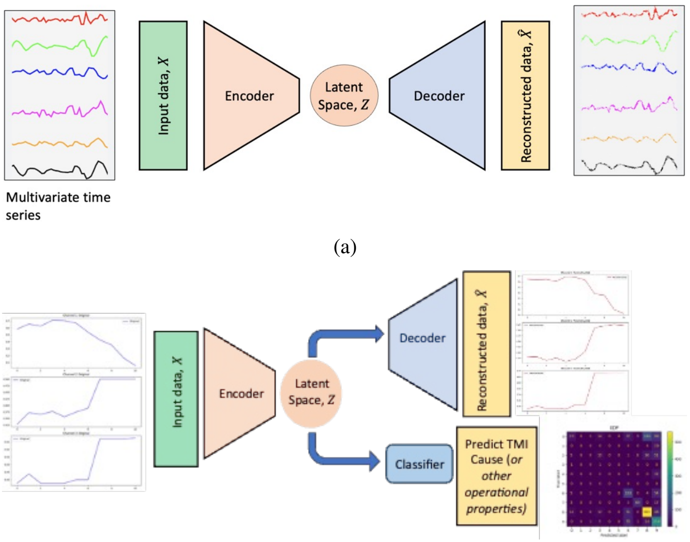
Fig. 1: Data structure with 2 hours of past data, 8 hours of forecasted data, and examples features pulled from METAR, TAF, and ASPM.

$$wind\_east = -V_s \times \cos{(V_d \times \frac{\pi}{180})} \tag{1}$$

$$wind\_north = -V_s \times \sin{(V_d \times \frac{\pi}{180})} \tag{2}$$

Then the preprocessed features are concatenated into a unified feature tensor $\mathbf{X} \in \mathbb{R}^{N \times D}$, where N represents the total number of hourly timesteps and D denotes the combined feature dimensionality, which is 14 for the features discussed. See Figure 1 for an example concatenation using 2 hours of past observations and 8 hours of forecasted events. During data preprocessing we store 24 hours of past data for each TMI event, but treat the number of past hours as a configurable training hyperparameter.

Finally, to enable supervised learning within our proposed representation learning architecture, we label each TMI event $\mathbf{X}$ with the recorded TMI-cause one-hot encoding vector $y \in \mathbb{N}^5$. Although there are 16 distinct TMI-causes recorded over the 2017-2025 data timespan, only 5 causes are populated to the one-hot encoding based on ATC feedback. The TMI-causes of interest include: wind, thunderstorms, snow/ice, low ceilings, and other. All TMI-causes not included in the first four causes are consolidated in the ”other” category. It is also possible to have two TMI-causes for a single TMI event if both a GS and GDP occurred. For conflicting causes, we defaulted to first to a weather cause and then fallback to the GDP cause if the conflict still persists. This consolidation helps focus on causes of most interest to ATCs and helps reduce class imbalance. Selecting TMI-cause instead of TMI-scenario was based on the intuition that TMI-causes encode more relevant operational information. Consider two TMI-events that share the same TMI-scenario, for example GDP→GS, but one event is caused by wind and the other low ceilings.


Multivariate time series
Input data, X
Encoder
Latent Space, Z
Decoder
Reconstructed data, $\hat{X}$
Classifier
Predict TMI Cause (or other operational properties)

(a)

(b)

Fig. 2: Comparison of autoencoder architectures: (a) Standard Variational Autoencoder with encoder and decoder components; (b) Modified Autoencoder with an additional classifier component enforcing contextual similarity in the latent space.

B. Representation Learning Framework

Neighbors-based methods are considered non-generalizing machine learning techniques, as they rely on 'remembering' all historical data rather than deriving abstract models. In contrast, Representation Learning aims to transform complex data into simpler, more interpretable forms, facilitating better understanding and analysis.

Our Representation Learning Model employs a deep learning framework, specifically utilizing a Variational Autoencoder (VAE) [7] and Long Short-Term Memory (LSTM) [4] hybrid architecture. This allows us to encode high-dimensional, sequential, historical data into a meaningful, low-dimensional latent space that preserves essential patterns and relationships.

Variational Autoencoder Architecture

VAE architectures are typically constructed using an encoder and decoder, which can be seen in Figure 2(a). The role of each component is as follows:

1. The encoder $E_\phi : \mathcal{X} \rightarrow \mathcal{Z}$ maps high-dimensional input data $x \in \mathcal{X}$ to a lower-dimensional latent representation $z \in \mathcal{Z}$, parametrized by ϕ.

2. The decoder $D_\theta : \mathcal{Z} \rightarrow \mathcal{X}$ reconstructs the original input from the latent representation, parametrized by θ.

The encoder outputs the parameters of a probability distribution (in our case Gaussian) from which the latent vector z is sampled:

$$q_\phi(z|x) = \mathcal{N}(z|\mu_\phi(x), \sigma^2_\phi(x)) \tag{3}$$

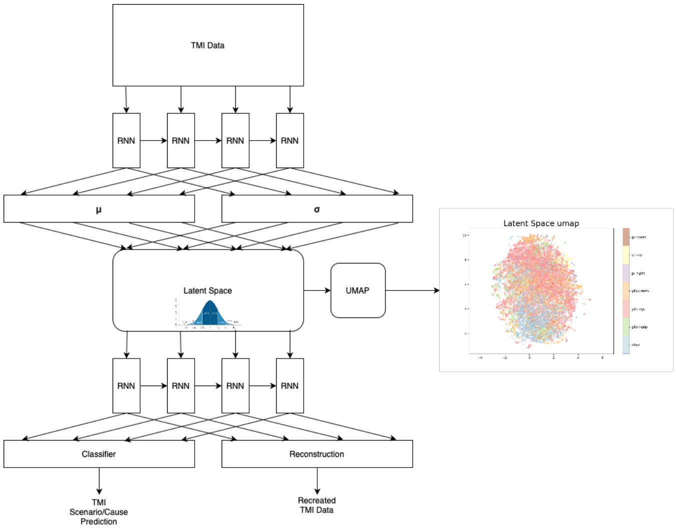


Fig. 3: LSTM VAE Architecture

This probabilistic mapping enables the model to capture uncertainty in the latent representation and helps smooth interpolation between separated classes in the latent space. Because of these characteristics, VAE models have been used previously for generative machine learning tasks by sampling from the latent space and using the reconstruction decoder for generation.

Classification Decoder

To enhance the abstraction capabilities of our model, we also incorporate supervised regularization that obligates the model to also predict the cause of the TMI. This approach helps differentiates between patterns by identifying whether a snapshot results in a Ground Delay Program (GDP) or a Ground Stop (GS) due to thunderstorm, wind, snow, or other factors.

Our modified architecture, which we call ALFRD, extends the standard VAE by integrating a second classification decoder, as shown in Figure 2(b). The classifier $C_\psi : \mathcal{Z} \rightarrow \mathcal{Y}$ maps latent representations to TMI cause classifications $y \in \mathcal{Y}$, parametrized by ψ. By training the second decoder simultaneously, the encoder is guided to map input data in a manner that both reconstructs the original data and preserves information required for TMI-cause prediction.

Multi-objective Loss

To further encourage TMI-cause differentiation in the latent space, we also included triplet margin loss as an optimization term during training. The total loss function is then formulated as:

$$L_{tot} = \alpha L_{recon} + \beta L_{KL} + \delta L_{pred} + \gamma L_{triplet} \tag{4}$$

Where:

- $L_{recon} = \frac{1}{n}\sum_{i=1}^{n} ||x_i - \hat{x}_i||^2$ represents the mean squared error between the original inputs x_i and their reconstructions $\hat{x}_i = D_\theta(z_i)$.

- $L_{KL} = D_{KL}(q_\phi(z|x)||p(z))$ denotes the Kullback-Leibler divergence between the learned latent distribution and a prior distribution $p(z)$, typically a standard normal distribution $\mathcal{N}(0, I)$.
- $L_{pred} = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(C_\psi(z_i))$ corresponds to the cross-entropy loss for predicting the TMI cause from the latent space.
- $L_{triplet} = \sum_{i=0}^{n}\sum_{p\in C}\sum_{n\notin C}\max(dist(a_i, p) - dist(a_i, n) + m, 0)$ where a_i represents a given anchor point within the the training set, p represents a positive point within the same class set C as a_i, n represents a negative point within a different class from a_i, and m represents an acceptable margin for the distance difference.
- α, β δ and γ are weights that balance the contributions of each optimization term.

By jointly optimizing for reconstruction accuracy, TMI-cause prediction, preserving a Gaussian distribution in the latent space, and TMI-cause differentation in the latent space, our model performs effective non-linear dimensionality reduction while preserving operationally relevant structures in the data.

LSTM Modifications to VAE Architecture

The encoders and decoders for VAEs are typically constructed using feed-forward neural networks, but because the TMI data has temporal structure, we opted to test architectures with inductive biases for sequential data. Specifically, we tested recurrent neural networks (RNNs), gated recurrent united (GRUs), LSTMs, and Transformers as mechanisms for encoding and decoding the TMI data. The best performing architecture was the LSTM and an example visualization of the LSTM enhanced VAE can be seen in Figure 3.

Each D dimensional hourly time-step from the length N sequence in the TMI data is passed as input into each LSTM cell. Then the output from each LSTM cell is then take as a lower-dimensional embedding $E < D$. This means the latent space $\mathcal{Z} \in \mathbb{R}^{N\times E}$ maintains the same sequential structure, while shrinking the data dimensionality at each time-step.

Training

During training, 6 months of data was withheld for model validation, and 1 year of data was withheld for model testing. Precisely, the training start-date is January 1st, 2017 and the training end-date is October 11th, 2023. The validation start-date is October 12th, 2023 and the validation end-date is April 12th, 2024. The testing start-date is April 13th, 2024 and the testing end-date is April 13th, 2025.

The LSTM enhanced VAE architecture was implemented and trained using the PyTorch [13] machine learning library. The encoder and reconstruction decoder are constructed using LSTMs with a hidden dimension of 64 and an output of 4 dimensional embeddings. The classification decoder is implemented as a two layer feed-forward neural network with rectified linear unit (ReLU) activations.

Gradient updates are performed with the multi-objective loss function from Eq. 4 using an Adam optimizer [6], a batch size of 256, and a learning rate of 0.0001 for 80 epochs. The optimization terms are weighted as follows: $\alpha = 10$, $\beta = 0.1$ $\delta = 10.0$ and $\gamma = 100.0$. Finally, we used 2 hours of past air traffic data and weather conditions during training, so each input tensor has dimension 11×14.

III. Results

We evaluated the efficacy of our representation learning approach using four methods: visual analysis of the latent space, classification decoder performance on the latent space, and k-nearest neighbor classification performance on the latent space. and k-nearest neighbor historical event retrieval. Each of these evaluations was performed using an entire year of data that was withheld from the model during training.

A. VAE Training

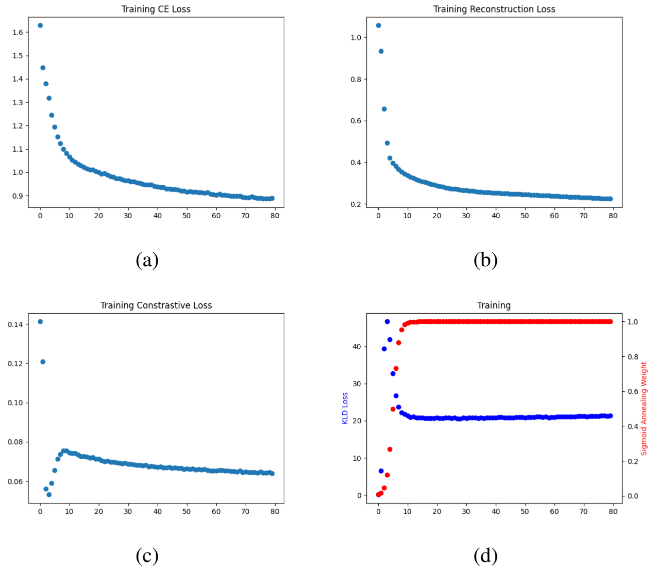
See Figure 4 for the training cross-entropy loss, training reconstruction loss, training contrastive loss, and training KLD loss. Similarly, see Figure 5 for the corresponding validation losses.

B. VAE Classification Testing Results

See Figure 6 for the VAE classification decoder testing confusion matrix. The model has an accuracy of 66.80%.

C. Latent Space Visualization

In our exploration of the latent space, we utilize the UMAP algorithm [10] to perform dimensionality reduction and visualization. In Figure 7, the UMAP algorithm is applied to the untransformed TMI testing data to generate a two dimensional embedding. While some TMI cause related structure is visible, the results are noticeably improved when we first use the



(a)

(b)

(c)

(d)

Fig. 4: VAE Training Losses

VAE to generate the latent space and then apply UMAP. The results in Figure 8 shows improved clustering and reveals underlying patterns that correspond to similar operational characteristics. These patterns demonstrate that points sharing similar causal factors are positioned in proximity within this reduced-dimensional space, significantly enhancing our ability to identify meaningful relationships.

D. KNN

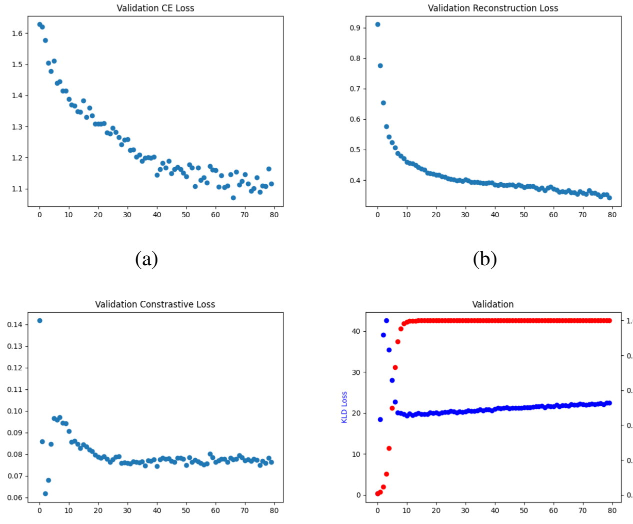
The KNN algorithm identifies similar historical events by minimizing a distance function between a query event q and historical events h_i:

$$\text{Nearest}_k(q) = \text{argmin}_k\{d(q, h_i) | h_i \in \mathcal{H}\} \tag{5}$$

where $d(\cdot, \cdot)$ represents a distance metric, $\mathcal{H}$ is the set of historical events, and argmin_k returns the k events with smallest distance values.

Our baseline implements a specialized KNN algorithm incorporating domain-specific distance metrics tailored for TMI operational data analysis. This approach combines a temporally-weighted Euclidean distance for continuous features—applying a non-linear weighting function that prioritizes current conditions (weight=1.0) while systematically discounting forecast accuracy with increasing temporal distance—with Jaccard distance for categorical features to account for discrete operational states.

See Table 1 for a comparison of KNN classifier testing accuracies using different data types and distance metrics. The custom KNN operating on the latent data achieves the highest accuracy of 65.03%. Then see Figure 9 for the custom KNN classifier's confusion matrix on the testing latent data.



(a)

(b)

(c)

(d)

Fig. 5: VAE Validation Losses

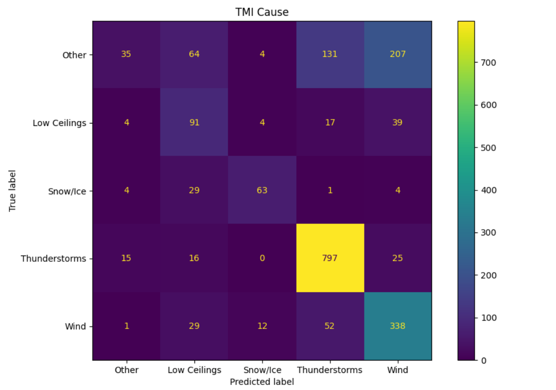


Fig. 6: VAE Testing Confusion Matrix

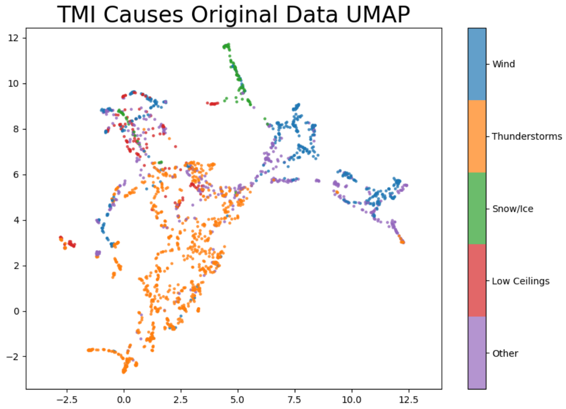


Fig. 7: Original Data UMAP

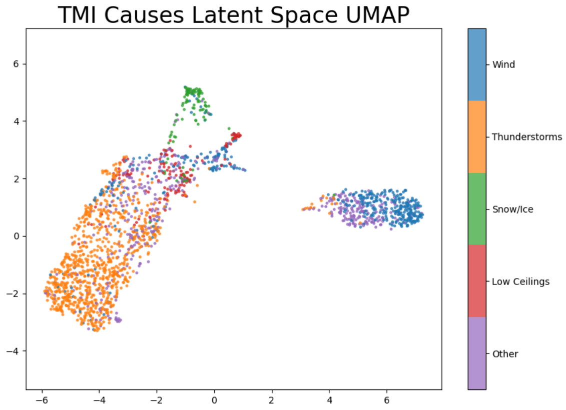


Fig. 8: VAE Testing Latent Space UMAP

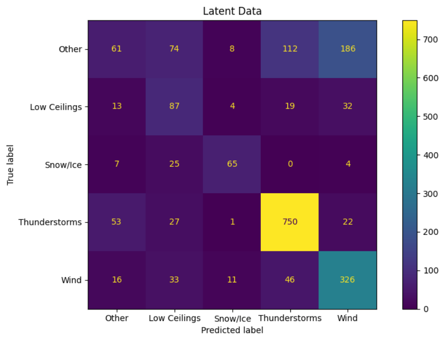


Fig. 9: KNN Latent Confusion Matrix

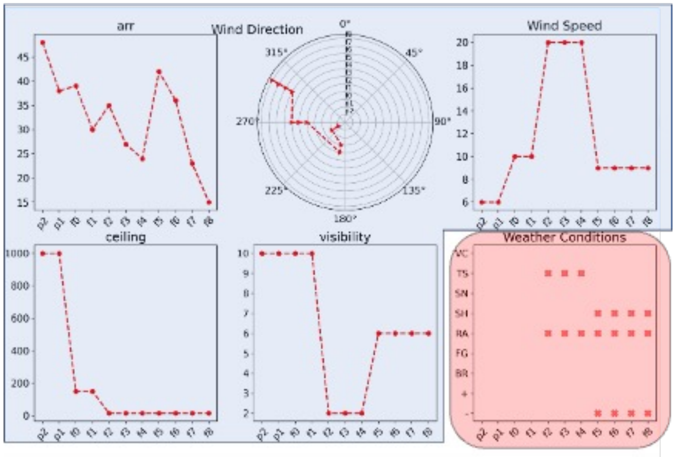


Fig. 10: Example of Raw Data: The blue area highlights time-series data (continuous feature), while the red indicates categorical features.

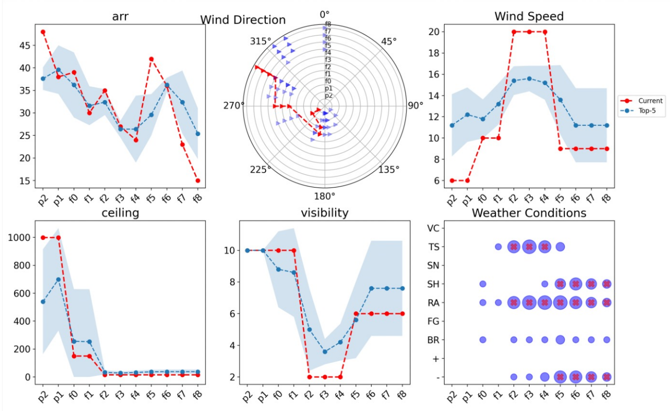


Fig. 11: Example of Similarity Analysis for a GDP caused by thunderstorm: The red line shows current scenario status. The blue line shows average of top-5 similar historical events. The shaded blue shows variance of top-5 similar historical events.

E. Historical Event Retrieval

Air Traffic Controllers frequently leverage historical precedents when making operational decisions. This practice of referencing similar past scenarios provides valuable context for current decision-making processes. We formalize this as a k-nearest neighbors (KNN) retrieval problem and explore various approaches to enhance retrieval quality.

Problem Context: Consider a Ground Delay Program (GDP) scenario caused by a thunderstorm, as depicted in Figure 10. In this situation, wind speed is forecasted to increase from 6 mph to 20 mph over the next three hours (f1, f2,..., f8 represent hourly forecasts), accompanied by a directional shift. Simultaneously, arrival rate capacity is expected to decrease significantly, with thunderstorm activity predicted, represented by the categorical feature 'TS'. To make informed traffic management decisions, controllers benefit from examining similar historical weather patterns and their operational outcomes.

Figure 11 demonstrates the efficacy of this approach by illustrating the centroid and variance of the five most similar historical events when applied to the exemplar scenario in Figure 10.

Representation Learning Approach: Our proposed enhancement transforms the complex, multi-dimensional TMI data into a structured latent space before applying standard KNN. This approach eliminates manual feature engineering and weighting, allowing the representation learning model to automatically identify and prioritize the most operationally relevant patterns.

KNN Results		
Data Type	Distance Metric	Accuracy
Latent *(ours)*	**Custom**	**65.03%**
Original	Custom	59.79%
Latent	Euclidean	61.81%
Original	Euclidean	53.38%
UMAP	Euclidean	49.70%
Latent	Dynamic Time Warping	61.96%
Original	Dynamic Time Warping	53.32%

TABLE I: Performance comparison of representation learning on TMI data using KNN. Our latent space method is benchmarked against PCA and UMAP, showcasing superior pattern recognition and efficiency.

We benchmark our representation learning approach against two additional dimensionality reduction techniques—Principal Component Analysis (PCA) and Uniform Manifold Approximation and Projection (UMAP)—each followed by standard KNN in the reduced space. Table 1 summarizes the comparative performance across all four approaches, demonstrating our method's superior ability to capture operationally meaningful similarities while maintaining computational efficiency.

IV. Conclusion and Future Work

This paper introduced an representation learning approach for identifying and retrieving historical Traffic Management Initiative (TMI) events to support air traffic management decision-making. By employing a specialized autoencoder architecture with joint optimization objectives, our methodology successfully transforms complex operational and meteorological data into meaningful latent representations that preserve critical temporal patterns and causal relationships. Experimental evaluation using data from New York Metroplex airports (JFK, EWR, LGA) demonstrates that our approach significantly outperforms traditional methods in scenario clustering and retrieval accuracy, providing air traffic managers with more precise and interpretable historical precedents.

The representation learning framework developed in this research systematically addresses the challenges of high-dimensional, heterogeneous TMI data while maintaining operational relevance. By incorporating classification-guided latent space organization, our model effectively disentangles complex causal factors underlying TMI events, enabling more nuanced similarity assessment than direct feature comparison. This capability proves particularly valuable for complex weather-related scenarios where subtle pattern recognition can substantially improve operational response.

Several promising directions for future research and development have been identified:

- **Conditional Variational Models:** We are exploring conditional variational autoencoder architectures to enable explicit conditioning of the latent space on specific operational parameters, further enhancing the precision of historical event retrieval for targeted operational contexts.
- **Cross-Airport Pattern Recognition:** Expanding our framework to identify and leverage similar weather-related patterns across geographically or operationally similar airports could enable transfer learning between facilities with limited historical data.
- **Human-in-the-Loop Evaluation:** Development of an interactive decision support interface that incorporates domain expert feedback to refine the similarity metrics and representation space based on operational utility and interpretability.
- **Temporal Attention Mechanisms:** Integration of attention-based architectures to better capture the relative importance of different time periods in the evolution of TMI events, potentially improving the model's focus on operationally critical moments.

Through ongoing refinement of these representation learning methods, we aim to develop a decision-support system that seamlessly integrates historical analogs, real-time operational data, and controller expertise to inform TMI deployment. Such a system has the potential to increase the efficiency, consistency, and effectiveness of strategic traffic management in the National Airspace System, especially under adverse conditions.

References

[1] Samantha J Corrado, Tejas G Puranik, Olivia J Pinon-Fischer, Dimitri Mavris, Rodrigo Rose, Jesse Williams, and Roohollah Heidary. Deep autoencoder for anomaly detection in terminal airspace operations. In *AIAA Aviation 2021 Forum*, page 2405, 2021.

[2] Zakaria Ezzahed, Antoine Chevrot, Christophe Hurter, and Xavier Olive. Bringing explainability to autoencoding neural networks encoding aircraft trajectories. In *13th SESAR Innovation Days 2023, SIDS 2023*, pages ISSN–0770. SESAR Joint Undertaking, 2023.

[3] Federal Aviation Administration (FAA). Faa aerospace forecasts: Fiscal years 2024–2044. https://www.faa.gov/sites/faa.gov/files/2023-FAA%20Aerospace%20Forecasts.pdf, 2023. Accessed: Apr. 24, 2025.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[5] V David Hopkin. *Human factors in air traffic control*. CRC Press, 2017.

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

[8] Xiaopeng Li and James She. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 305–314, 2017.

[9] Farzan Masrour Shalmani, Milad Memarzadeh, Aida Sharif Rohani, and Krishna Kalyanam. Ground stop adjuster: A machine learning approach to improve air traffic management initiatives. In *AIAA SCITECH 2025 Forum*, page 2121, 2025.

[10] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[11] Milad Memarzadeh, Bryan Matthews, and Ilya Avrekh. Unsupervised anomaly detection in flight data using convolutional variational autoencoder. *Aerospace*, 7(8):115, 2020.

[12] Xavier Olive, Junzi Sun, Adrien Lafage, and Luis Basora. Detecting events in aircraft trajectories: rule-based and data-driven approaches. In *Proceedings*, volume 59, page 8. MDPI, 2020.

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA, 2019.