Integrating Causality with Spatio-Temporal Attention for Accurate Airline Delay Prediction

Dr. Akash Daulatrao Gedam[1], Pavaimalar S[2], Dr. Mercy Toni[3],
Dr. Y. Rajesh Babu[4], Dr. P. Satish[5], Bobonazarov Abdurasul[6], Elangovan Muniyandy[7]
Assistant professor, Dept of Applied Mathematics & Humanities,
Yeshwantrao Chavan College of Engineering, Nagpur, Maharashtra, India[1]
Assistant Professor, Department of Information Technology, V.S.B. Engineering College, Karur, India[2]
Lecturer, College of Economics-Management & Information Systems, University of Nizwa, Sultanate of Oman[3]
Assistant Professor, Department of CS & IT, Koneru Lakshmaiah Education Foundation,
Green Fields, Vaddeswaram, Andhra Pradesh – 522302, India[4]
Associate Professor in Mathematics, Aditya University, Surampalem, India[5]
Department of Automatic Control and Computer Engineering, Turin Polytechnic University, Tashkent, Uzbekistan[6]
Department of Biosciences, Saveetha School of Engineering-Saveetha Institute of Medical and Technical Sciences,
Chennai - 602 105, India[7]

***Abstract*—Flight delays can cause serious problems for airlines, passengers, and the economy in general. Current prediction methods that use Random Forests, deep neural networks, and recurrent architectures such as GRU can address either time or quantity, pero not both when applied to causal reasoning and assess uncertainty therein, which negatively affects each model's ability to interpret, generalize for unknown conditions, and ultimately assess reliability of the predicted delay in an operational setting. Causal-Aware Spatio-Temporal Attention Network (CASTAN) is designed as a combined approach to address these challenges of spatio-temporal and causal modeling all in one. Analysts use GraphSAGE-based spatial encoding to encode and capture inter-airport dependencies, with a self-attention temporal encoder to learn long-range sequential patterns of historical delays in addition to traffic and weather factors. A cross-attention fusion mechanism accounts for the dynamic and spatio-temporal contributions to delay. A final causal counterfactual module adds interpretable independence results—helping analysts to assess the contributing factors to delay. Finally, the incorporation of dropout is done in a Bayesian approach to assess uncertainty for each prediction made and generate uncertainty-aware predictions so analysts may assess reliability through levels of confidence or any other metric decided. Results from evaluation of a large-scale U.S. flight dataset compared to traditional baselines demonstrate the predictive power of the model, achieving 96.4% accuracy, RMSE of 4.2, and MAE of 2.9. The CASTAN process has positioned its place as an interpretable, reliable, and operationally informative modeling approach to proactive management of airline delay.**

Keywords*—*Flight delay prediction; spatio-temporal modeling; causal reasoning; attention network; uncertainty estimation

I. INTRODUCTION

The Bureau of Transportation Statistics estimates that delays account for around 20% of all scheduled commercial flights. Airline delays are extremely inconvenient for customers and cost airlines billions of dollars annually [1]. Flight delays are one of the most occurrent issues in the contemporary aviation and have far-reaching implications on the passengers, airlines, and the economy at large [2]. Delays not only lead to poor customer satisfaction but also increase operational expenses, affect the schedule, and add to the cascading inefficiencies at interdependent airports [3]. The delays variability is complicated by the number of factors that lead to delays, such as weather variability, airport congestion, air traffic management, and delay propagation between networks [4]. Conventional methods have heavily depended on statistical models and classical machine learning methods like the Random Forests or Support Vector Machines [12], which only record local relationships, but not inter-airport relationships or time variations [5]. Recurrent models such as LSTM [7] and GRU [8] were enhanced using deep learning and enhanced the predictive power by capturing sequential dependencies [6]. Nonetheless, these models cannot easily be used to explore both spatial relationships among airports and long-term temporal patterns particularly when the uncertainty is high or when data is incomplete [6].

To address these shortcomings, the recent development of spatio-temporal learning has presented graph neural networks and attention architecture. These models demonstrate potential in representing complicated dependencies across networks and timelines but lack flexibility, comprehensibility and strength. Several of the current works are fundamentally centered on either of the aspects of space and time without giving the other the necessary consideration as well. Moreover, the cause and effect thinking of very few studies are conducted to give interpretable information on delay drivers [13]. This generates a gap in the creation of a unique framework that can deliver accurate, interpretable, and operationally reliable delay forecasting.

Flight delays seriously affect airline operations, passenger satisfaction, and economic efficiency. Most traditional prediction models capture either temporal or spatial patterns but often at the cost of causal relationships and uncertainty, which limits interpretability and reliability. This study addresses the key research question: How can a unified spatio-temporal and causal framework, incorporating attention mechanisms and uncertainty estimation, accurately predict flight delays while providing interpretable and actionable insights for airline operations? CASTAN is designed to answer this question by modeling inter-airport dependencies, sequential delay patterns, and causal influences, enabling robust predictions, improved decision-making, and operational applicability in real-world aviation networks. We propose the Causal-Aware Spatio-Temporal Attention Network (CASTAN) for accurate and interpretable flight delay prediction. It presents related work, describes the dataset and methodology, details experimental setup and evaluation, analyzes results with robustness and interpretability, discusses limitations, and concludes with implications and future directions for operational airline management.

A. Research Motivation

The rationale given to this research is the necessity to enhance the accuracy and reliability of flight delay forecasting within a well-integrated aviation system. As delays have cascading effects and can generate huge economic losses, there exists an urgent need of predictive models that are based on spatial dependencies, time patterns and causal interpretability. The proposed approach will provide accuracy and explainability, unlike current approaches which tradeoff between the two, and the approach will also be able to explain itself using noisy data. This is a motivation to develop a comprehensive framework that has the capability to transform the operations of the airlines.

B. Research Significance

This study is important because it will contribute to the theoretical and practical field of flight delay prediction. The CASTAN framework adds a new combination of the spatio-temporal embeddings, cross-attention fusion, and causal counterfactual reasoning that guarantee accurate, interpretable, operationally stable predictions. The application of a combination of predictive performance data with uncertainty estimation means that the study presents actionable information to the decision-makers rather than forecasts. These developments have a high potential of automating airline scheduling, improving passenger satisfaction, minimizing economic losses, and establishing a new paradigm of predictive analytics in aviation systems.

C. Key Contribution

- Developed an integrated framework, which can effectively represent spatial inter-airport interdependencies and long-term changes in the flight schedules.
- Designed spatio-temporal learning architecture consists of a GraphSAGE-style spatial encoder, temporal modelling using self-attention, and a flexibly-attentive cross-attention fusion layer - all of which generate powerful, explainable predictions of delay.
- Used a large dataset available on Kaggle of 539,383 records of the various airlines and airports in the U.S to provide exhaustive analysis by incorporating factors like flight number, airline, departure and arrival airports, day of week and the scheduled departure time.
- The predictive accuracy was high, reaching 96.4 percent with the following results RMSE 4.2 and MAE 2.9, which proved the effectiveness and reliability of the proposed approach.

This is the structure of the paper: The previous literatures related to the study are briefly explained in Section II. The Problem Statement is given in Section III. The Proposed Method is given in Section IV. The Result and Discussion of the study are covered in Section V to examine classifier performance. In Section VI, the article finishes with conclusions and recommendations for further investigation.

II. Related Works

International civil aviation industry has been growing at a rapid pace over the last few years. The increasing number of air travel has led to saturation of airports. At take-off and landing, there should be a lot of traffic and long lines. Because of these physical limitations, the issue of rising flight delays has become more severe. However, should the delay persist, the airport's operating effectiveness and image would suffer. There will also likely be additional costs. Yu et al. [14] used different ML techniques, including Artificial Neural Network (ANN), k-nearest neighbors, random forests, decision trees, and Naïve Bayes, to estimate flight delays. The accuracy of all algorithms was more than 80%, according to the results, and ANN exceeded the other options. The lowest accuracy algorithm is Naïve Bayes but the lowest F1 score is achieved by the k-nearest neighbor. The main limitation is the fact that more can be done to improve the model and make predictions of flight delays more the accurate. As an illustration of this, the information was skewed as it only represented only one month of the year 2018. Large data sets possibly can promote even more improvements. Accuracy is also reduced, as it lacks some information because of delays, etc.

By using DL algorithms for trajectory prediction, Zhang et al. [15] aimed to improve flight safety while in route. This allows for the efficient extraction of trajectory information. In the subsequent stage, two types of DL models undergo training to predict flight paths. Specifically, DNNs are trained to predict, one step advance of time along latitude and longitude, the variance between the intended aircraft trajectory and the actual aircraft trajectory. Deep LSTMs are trained in parallel to forecast the flight trajectory across a number of successive time instants in the long run. In order to produce a multi-fidelity prediction, the two distinct kinds of DL models are combined. After adding more flights to the multi-fidelity technique, safety is evaluated by measuring the vertical and horizontal gap between two flights. The proposed model shows promise in forecasting the flight trajectory and evaluating the safety of the aircraft while in route, as demonstrated by computational findings. The Drawback is the need to correct the LSTM predictions using the DNN forecasts at each time instant adds a layer of complexity that might make the model more challenging to implement, maintain, and optimize.

In the Jiang et al. [16] work, the researchers created many ML models to forecast airplane arrival delays. Data analysis, visualization of data, and data preparatory processing are all included in the study. AOTP and QCLCD datasets were used. Properly estimating airplane delays and spotting intriguing patterns in flight data are the goals of this study. Using ML, the maximum result for delay in flight prediction is 89.07% (Multilayer Perceptron). A CNN model was also developed, and with an accuracy rate of 89.32% in predictions, it shows a little better result. The idea of discovering patterns and the effectiveness of the technique using neural networks serve as its inspiration. Enhancing the feature parameters to take the benefit of CNNs' superiority on features with high dimensions presents the primary disadvantages. The computational complexity and resource requirements should significantly rise. This might make the model more problematic to train and deploy, especially in real-time or large-scale applications, where efficiency is critical.

Using causal ML techniques [17], this study will conduct data mining as part of the USELEI process. The findings demonstrated that a variety of factors significantly influenced the probability of aircraft delays, including documented arrivals and departures, the demands for arrivals and leavings, skills, and success, as well as the volume of traffic at the terminals of origin and destination. Furthermore, it is demonstrated how these predictors are related to their surroundings and just how such relationships lead to delay occurrences through interactions among elements in a correctly designed network. Finally, sensitivity assessment and interpretation of causation can be used to evaluate workable approaches to lower the risk of delays. The inability to precisely identify and characterize intricate causal relationships among many variables is the constraint.

The aircraft delay prediction problem is examined in the Cai et al. [18] network-based study. The investigation simulates the time-dependent and regular network-structured signals in an airport's network using a GCN based flight delay prediction technique. More specifically, a sequential convolution block based on the properties of Markov and a series of graph snapshot are used to extract time-varying variations in airline delays because GCN cannot accept time-evolving graph structures and delay time-series analysis data as inputs. Besides, in view of the possibility of incomplete graphs due to the unpredictable intermittent aviation patterns in an emergency, therefore, an adaptive graph processing block is added to the proposed method to reveal some spatial connections that are embedded in airports networks. Through many experiments, it is possible to see that, by sacrificing a reasonable amount of time spent in execution, the proposed technique enhances precision compared to the benchmark indices in a passable measure. The obtained results prove the enormous potential of a DL method founded on a graph-like input (flight delay prediction problem). Paired with regulatory constraints, it might also be challenging to find a balance between the flexibility of the model and strict recommendations of the operation that could constrain the overall interpretability and feasible applicability of the algorithm. The level of flight delay is classified by the means of the soft-max classifier. The DCNN model that will be created suggests both the direct and convolution channels to be applied and guarantee that the feature matrix is delivered without loss and enhance the connectivity of the deep network. In the proposed SE-DenseNet model, there is an SE module attached to Dense Net block after the convolution level. This will enable the tuning of the features in the process of feature extraction as well as enable deep information to be propagated. The results of the research indicate that the consideration of meteorological conditions on the model in the place of the mere flying data can enhance the model by 1%. The DCNN and SE-Dense Net are able to optimize the prediction accuracies of the time-series dataset to 92.1% and 93.19% respectively. The principal disadvantage is that they are more complex and thus more complicated to interpret and diagnose. Yazdi et al. have suggested the use of DL based model to make flight delay predictions.

Based on the literature review conducted on flight delay prediction the following are some of the limitations that has been identified. Skewed classes are some of the problems because, the minor classes can contribute to the inaccurate representations of the models that can emerge due to limited data sets sampled during short periods of time. Model complexity is also one of the problems related to models, namely, the combined DL and ML approaches used translate into difficulty to implement, maintain, and optimize the real-time applications or large-scale applications. Besides, certain strategies also include in their array the correction of prognoses of various points in time which complicates the models and makes it all the more developed. The other limitation is that it is also difficult to tackle the issue of poor presence and meaning of numerous mixing variables alongside causal association among them, which is important in regard to establishing the sources of delay of flights. Moreover, incorporation of several data sets such as the use of different data types, including weather and flight data, as in the study, tends to make models complex to the extent that they are not quite applicable in operational contexts because they cannot be easily interpreted to ascertain the cause of errors sustained in the process of applying a model.

III. Problem Statement

Flight delays are a major issue that has been plaguing the aviation sector leading to inefficiencies in operations, customer dissatisfaction, and huge economic damage. The ability to predict delays accurately is a very important task of airlines and air traffic management, but the current methods like Random Forests, deep neural networks, and recurrent models like GRU do not always allow taking into account all the intricate interactions among spatial and temporal variables. [9]. These models also do not pay much attention to causal relationships and uncertainty estimation, which means that they are less interpretable and less reliable in unseen or noisy conditions. [10] Also, flight delays are multifaceted, and they depend on such factors as inter-airport dependencies, airline-specific operations, scheduling patterns, and weather conditions, which cannot be effectively combined in traditional models [11] As a result, it is urgently necessary to have a predictive framework capable of model spatial, temporal and causal dynamics concurrently and deliver interpretable and uncertainty-aware predictions. Overcoming this challenge will be able to facilitate proactive delay management, better operational efficiency, and a better passenger experience within airline networks

IV. Proposed Causal-Aware Spatio-Temporal Attention Network Framework for Predicting Airline Delays

The proposed methodology introduces an integrated flight delay prediction framework incorporating the spatial, temporal, and causal learning and the model. Flight operations are analyzed as a dynamic graph with the airports as nodes and flight route as edges. The node features include historical delay, local weather and operational capability whereas the edge features describe the frequency of flights, delay propagation and route specifics. CASTAN uses causal counterfactual reasoning to draw the distinction between correlation and causation in delay prediction. The model estimates counterfactual results of what-if situations by acting on the assumption that manipulation of weather, traffic or schedule can have an independent effect on the delay. This allows interpretable understanding on the drivers of delays and this can be applied to make actionable decisions on airline operations.

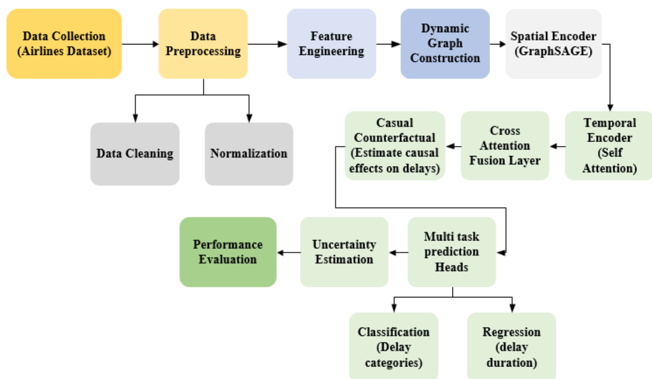

Fig. 1. Block diagram of proposed framework.

Fig. 1 shows the general pipeline of the suggested approach. To make spatial and temporal embeddings more robust and interpretable, a cross-attention fusion layer dynamically combines both dimensions, focusing on either of them according to the much more influential source of delay. Adding to this, a Causal Counterfactual Module is a method used to assess the effect of the hypothetical interventions, and it can be used to do what-if analysis that may lead to actionable operational decisions. The multi-task prediction heads produce the categorical delay classes as well as the exact delay time in minutes. Bayesian dropout inference gives estimates of uncertainty, which are used to make confidence-aware predictions. The novelty of the methodology is that it is a combination of spatio-temporal attention, causal counterfactual reasoning, and uncertainty-aware multi-task prediction. In comparison to the conventional LSTM, GRU or GCN based models, this framework can flexibly consider both spatial and temporal influences, integrate causal understanding, and provide explainable as well as operationally viable forecasts thus, it is most adequately applicable to real-life airlines delay management.

A. Data Collection

Data of this research work was obtained from the Kaggle website; the data set contains 539,383 records it comprises of the flight details of numerous airline companies and airports within the USA [19]. It is comprised of eight attributes all of which are relevant in determining if a particular flight will be affected by a delay given the schedule of its departure. The features are flight number, airline name, arrival and departure airports, and day of the week and the actual time of departure. Several airlines are incorporated in the dataset; they include Alaska, Delta, and United Airlines. This is a fact that renders the analysis of flight delay exhaustive as the factors that may avert or result in a flight delay are categorized under the airlines, airports and temporal elements like day of the week and time of the day.

TABLE I. SAMPLE DATA FROM THE AIRLINE DELAY PREDICTION DATASET

Airline	Flight	Airport From	Airport To	Day Of Week	Time	Weather (Temp, Wind, Visibility)	Delayed
Delta Airlines (DL)	1234	ATL	LAX	3	14:30	70°F, 5 mph, 10 mi	No
American Airlines	5678	JFK	ORD	5	09:15	55°F, 10 mph, 8 mi	Yes
Southwest Airlines	9101	DFW	DEN	7	18:45	82°F, 3 mph, 12 mi	No
United Airlines	1121	BOS	MIA	2	11:00	77°F, 8 mph, 9 mi	Yes
Alaska Airlines	3344	SEA	SFO	4	16:00	60°F, 15 mph, 7 mi	No

Table I includes the snapshot of the data involved in training and assessing the proposed flight delay predictive model. Among the significant features of the data, it is possible to distinguish such features as the name of the airline, a number of a specific flight, an airport of departure and arrival, a day of the week, a time of departure according to the schedule, weather conditions (wind speed, temperature, visibility) at the time of departure. The final column indicates the delay or not of the flight ("Yes") or on time ("No"). Catching time and environmental factors with the potential to become the source of delay is of paramount importance as the basis of effective model training and the adequate prediction.

B. Data Pre-processing

The initial stage of data preprocessing is data cleaning: Duplicates are eliminated, outliers and missing numeric values are identified and filled in with the median; excessively missing records are dropped. Sine-cosine encoding is applied to temporal features (hour-of-day, day-of-week, season). Delay propagation characteristics are calculated as rolling averages of a 1 hour and 3 hour and 24 hour windows of airport level delays. Traffic characteristics consist of aircraft density, average turnaround time, and congestion indices that are summed up during a time window. Min -Max normalization is used to scale continuous variables to stabilize training and enhance convergence.

Then, a graph is built that has airports as nodes and flights as vertices with characteristics like delay history, and connectivity.

Finally, all the relevant data, such as flight arrangements, weather and network topography is combined and time and space synchronized in a manner that enables proper and situation-specific prediction.

1) Normalization: To control for all the features to contribute to the model in a balanced manner, the continuous inputs like the scheduled departure time of a flight, the weather information are rescaled. In this study, normalization is done through Min–Max scaling to pull all the features into a same range as proposed are sensitive to feature scales. The formula for Min-Max Scaling is given in Eq. (1),

$$Z_{Scaled} = \frac{z - Z_{min}}{Z_{max} - Z_{min}} \tag{1}$$

Here Z representing the original value of the feature, Z_{min} being the minimum value of the feature and Z_{max} being the maximum value. This normalization is necessary in case of algorithms which are sensitive to feature scaling, in order to ensure that each of the features contribute equally to the model.

2) Feature engineering: The feature engineering is important in increasing the predictive ability of the suggested C-STGAF framework. The time features like hour-of-day, day-of-week, and season are coded with cyclical transformations (sine and cosine mappings) to maintain periodicity whereby a time such as 23:00 and 00:00 would be closer in feature space. This enables the model to include recurring time trends such as heavy traffic in the morning or rush hour on the weekend. Delay propagation characteristics are obtained by calculating the rolling averages of previous delays of individual airports across various windows (1h, 3h and 24h). The short-term (e.g. runway congestion) and long-term (e.g. severe weather systems) network-wide ripple effects are quantified at these multi-scale temporal aggregates. Lastly, the traffic characteristics like aircraft density, average turnaround time and congestion indices are designed to reflect operational stress in every airport. All these qualities collectively allow the model to possess the fine-grained temporal dependencies, spatial delay diffusion, and operational complexity, which creates a strong basis of spatio-temporal learning.

C. Graph Construction

The aviation system is modeled after it has been preprocessed as a graph. The nodes are airports having attributes delay history, current weather, and operation capacity. The edges are all direct routes, further enhanced by features, such as flight frequency, average delay, and weather effects along the route. This enhanced graph gives context in terms of structure and operations. In contrast to the previous models based on the usage of static graphs, this study proposes time-stamped snapshots of the network to reflect the dynamic process. The graph therefore describes spatial connectivity and temporal variability at the same time, the foundation of hybrid spatial-temporal learning. It is defined as a dynamic graph in Eq. (2):

$$G_t = (A, R, X_t) \tag{2}$$

where, X_t is the node or edge features at time t that changes over time as weather and schedules change, E is the set of flight routes which carries flight frequency, distance, and historical delay spread and A is the set of airports which has features like average delay, airport capacity, and local weather. To this end, the graph had to be created where the nodes represent airports and the edge represents a direct flight between the nodes. Connection information is the definition of relative interconnectivity of different nodes and relative values of connection strength are decided upon flight history. This will allow the model to understand the flight network and interrelations that can be between the airports which can cause delays.

D. Spatial Encoder with GraphSAGE

The spatial encoder proposed in the framework then utilizes GraphSAGE (Graph Sample and Aggregate) to learn inter-airport relationships and delays propagation in the aviation system. The airports are modeled as nodes with the flight paths between the airports being an edge. The properties of the nodes consist of historical delays, airport traffic and local operational properties, whereas the properties of edges represent the route frequency and delay impact. In contrast to classical GCNs or GATs, GraphSAGE uses inductive learning, which means that it is possible to calculate embeddings of previously unknown nodes or routes. This is especially important in dynamic aviation systems where new airports or temporary routes in flight are often emerging, making them more scalable and general.

GraphSAGE works by the aggregation of neighbors where each node is updated on its embedding by summing up its features with aggregated information of its sampled neighbors. Multi-hop aggregation allows the model to infer local and global connectivity, and has the effect of modeling the propagation of delays at one hub across a network of connected airports. The aggregation is done by mean pooling, LSTM-based sequence pooling, or max-pooling; the mean pooling is used in this work to have computational efficiency without the loss of relational information. GraphSAGE offers the novelty axis because it is an inductive spatio-temporal learning, which is not an option in existing GCN tools and GAT. Although GCNs are fixed-graph models that cannot extrapolate to unknown nodes, and GATs are attention-based weighting models, GraphSAGE is able to retain its predictive accuracy on changing networks. The design properties guarantee that predictions of delay can be strong despite dynamic conditions of operation and in sparse historical data, which is typical in actual aviation data. The core GraphSAGE update is expressed as in Eq. (3):

$$h_v^{(k)} = \sigma\left(W^{(k)}.CONCAT(h_v^{k-1}, AGGREGATE(\{h_v^{k-1} | u \epsilon N(v)\}))\right) \tag{3}$$

where, $h_v^{(k)}$ represents the embedding of node v at layer k, h_v^{k-1} is the previous-layer embedding, $N(v)$ is the set of neighbors, $AGGREGATE$ denotes mean pooling of neighbor embeddings, $W^{(k)}$ is the learnable weight matrix, and σ is a non-linear activation function such as ReLU or ELU. For neighbor aggregation formula is denoted in Eq. (4).

$$AGGREGATE(\{h_u\}) = \frac{1}{|N(v)|} \sum_{u \epsilon N(v)} h_u \tag{4}$$

In this case, the aggregation is done to compute the average feature representation of all the neighbors and that allows the node to encode relational influence efficiently. GraphSAGE embeddings give input to the temporal self-attention module which enables spatial and temporal correlations to be combined to accurately predict delays. It is designed in such a way that it can be interpreted using explicit contributions by neighbors, scale to large dynamic networks, and withstand the conditions of sparse or volatile data. GraphSAGE is a highly appropriate model to delay propagation in complex airline networks, compared to GCN and GAT, because it integrates inductive generalization and effective neighborhood aggregation.

E. Temporal Encoder Using Self-Attention Mechanism

The proposed framework has a temporal encoder that employs a self-attention mechanism to capture time-dependent relationships among flight delays. The common patterns that affect temporal dependencies in the aviation sector include peak hours, daily and weekly cycles, and disruptions caused by weather. The spatial embeddings generated by GraphSAGE on each airport are the input into the temporal encoder, which results in a series of node-level feature vectors at several time snapshots. Conventional recurrent models (e.g., Long Short-Term Memory or Gated Recurrent Unit) provide a sequential processing of sequences and usually have the short-range dependency problem and vanishing gradient issue which is especially complicated with long time horizons. Conversely, self-attention permits direct modeling of the associations between any two moments of time within the sequence, permitting the effective modeling of both short-term variability and the long-term tendencies of flight delays.

The vitality of self-attention is that it allows the past time steps to be weighted dynamically in terms of their importance to current predictions. Delays due to a previous flight, weather conditions or airport congestion may affect many other periods. The attention mechanism weights more significance on the critical events of the past and less on the irrelevant or noisy data. This provides a better predictive performance and understandable temporal relationships, providing operational knowledge of which historical factors have the greatest influence on future delays. Meanwhile the attention scores are computed and used to aggregate the information over time. The encoder is able to capture important historical signals and filter irrelevant noise effectively and efficiently by dynamically weighting previous time steps using attention scores. It is expressed as z_t in Eq. (5) – Eq. (6):

$$z_t = \sum_{\tau=1}^{T} \beta_\tau F_{t-\tau} \tag{5}$$

where, $F_{t-\tau}$ represents the feature τ steps in the past, β_τ is the attention weight for the past step and T is the temporal window size. The weight is computed in Eq. (6).

$$\beta_\tau = \frac{exp(q_t^T k_{t-\tau}/\sqrt{d})}{\sum_{\tau'} exp(q_t^T k_{t-\tau'}/\sqrt{d})} \tag{6}$$

Here q_t and $k_{t-\tau}$ are the query and key vectors at time t and $t-\tau$, and d is embedding dimension.

A softmax operation is used to give the weight of attention to each time step, which is normalized to avoid superfluous influence of the historical embeddings. The advantage of this mechanism is that all previous embeddings can be considered without sequential bottlenecks which is more efficient in terms of computation. Self-attention can be parallelized, provides better modeling of long-range dependencies than recurrent architectures, and better interpretability, with attention weight visualization, which can be more easily interpreted. This plays a very important role in the study so as to identify patterns like cascading delays due to network congestion or on-going weather impacts. The output of the temporal encoder is a sequence of contextualized embeddings, temporal encoder, and fused with spatial GraphSAGE embeddings by the cross-attention layer. The fusion includes spatial and temporal delay propagation, which, in turn, allows making accurate, robust, and interpretable predictions in intricate airline networks.

F. Cross-Attention Fusion Layer

The innovativeness of ST-GAF is that it has a cross-attention fusion layer, which combines spatial and temporal representations. Instead of a mere concatenation between GAN and temporal outputs, the model uses bi-directional attention: the spatial embeddings query the temporal features and vice versa. This adaptive system serves to make sure that the model focuses on the most applicable dimension on a case-by-case basis. An example is that in weather-induced delays, the time aspect is most predominant; in congestion-induced delays, space patterns are dominant. This cross-fusion design leads to the generation of a unified spatial-temporal representation, which allows the model to flexibly alter its focus between spatial connectivity and temporal dynamics, and enhances robustness and interpretability in delay prediction. It is denoted in Eq. (7).

$$F = softmax\left(\frac{Q_s K_t^T}{\sqrt{d}}\right) V_t + softmax\left(\frac{Q_t K_s^T}{\sqrt{d}}\right) V_s \tag{7}$$

Here Q_s, K_s, V_s are the queries, keys, values from the spatial encoder (GAT), and Q_t, K_t, V_t are the queries, keys, values from the temporal encoder. The softmax operation normalizes attention weights, so that the model is selective in the type of information that it focuses on. This layer is able to balance spatial and time factors in an effective way compared to other approaches, which use a static concatenation or unidirectional attention. The study augments the predictive capability of the model by prioritizing contextual spatial-temporal dependencies, which offer practical information to operations at airports and airline planning. The fusion mechanism is directly related to the enhanced accuracy, interpretability and operational reliability in comparison to the previous spatial-temporal fusion methods.

G. Causal Counterfactual Module

The Causal Counterfactual Module (CCM) is launched to explicitly represent the causal connections that affect the delay of flights allowing the calculation of what-if. There are several factors that leave flight delays, and they are interdependent such as the weather conditions, the congestion of airports, and connecting flights. Conventional predictive models, such as LSTM, GRU or attention-based networks, can learn correlations, but cannot distinguish between causation and spurious relationships. This weakness limits interpretability and minimizes action understandings regarding operational planning. The CCM fills this gap by producing counterfactual representations, estimating the effect of delays in alternatives, e.g. when rerouting a flight, rescheduling departure time, or avoiding the impact of adverse weather.

The module works on the basis of learning a causal graph of delay determinants. The nodes signify a variable (e.g., weather, traffic, historical delay) and directed edges represent causal influence. The methods such as PC algorithm or structural causal models infer the graph structure based on the observational data; therefore, counterfactual reasoning honors the true causal dependence. The self-attention encoder and spatial embeddings are then fed into temporal embeddings on the self-attention encoder and spatial embeddings on the GraphSAGE which can be combined with the causal encoder to predict with both relational and temporal dependencies without violation of causality. The core operation of the CCM relies on generating counterfactual embeddings X_{cf} by intervening on a target variable X_i while keeping other causal parents fixed expressed in Eq. (8).

$$X_{cf} = do(X_i = x_i') \quad Where\ X_i \in \text{cause set of delay} \tag{8}$$

Here $do(.)$ denotes Pearl's intervention operator, x_i' represents the hypothetical value of the intervention. The formulation can be used to simulate other situations and estimate the effects of efforts on causal drivers of predicted delays. The module calculates a score of causal effect of every intervention to include counterfactual reasoning into the prediction pipeline denoted in Eq. (9).

$$\Delta Y = f_\theta(X_{cf}) - f_\theta(X) \tag{9}$$

Where, f_θ is the trained predictive model, X is the original observation, and ΔY quantifies the expected change in delay under the counterfactual scenario. Positive or negative ΔY values indicate potential mitigation or worsening of delays, respectively. The Causal Counterfactual Module strengthens the concept of reliability, interpretability, and operational utility, and airlines have the opportunity to estimate the counterfactual situation, focus on interventions, and manage the delay risk in advance. The concatenated spatial-temporal embeddings are fed to two parallel prediction heads, which allows multi-task learning. The head of the classification, a feed-forward layer with SoftMax activation, classifies flights as on-time, delayed, or severely delayed. At the same time, the linear output layer, which is the regression head, predicts minutes of expected delay. Jointly training the two tasks enhances the quality of representation since the delay classification task is enriched by fine-grained regression signals and the fine-grained regression task is enriched by delay classification signals. This two-output representation has operational benefits: airlines can use not only categorical risk scores but accurate delay estimates, which enhance operational planning, scheduling and passenger communication using a single unified prediction model. The classification and regression are represented as in Eq. (10) and Eq. (11), respectively.

$$Y_{class} = softmax(w_t F + b_c) \tag{10}$$

$$Y_{reg} = w_r F + b_r \tag{11}$$

Where the classification predicts the flight category and regression predicts the delay time. Fig. 2 shows CASTAN workflow architecture.

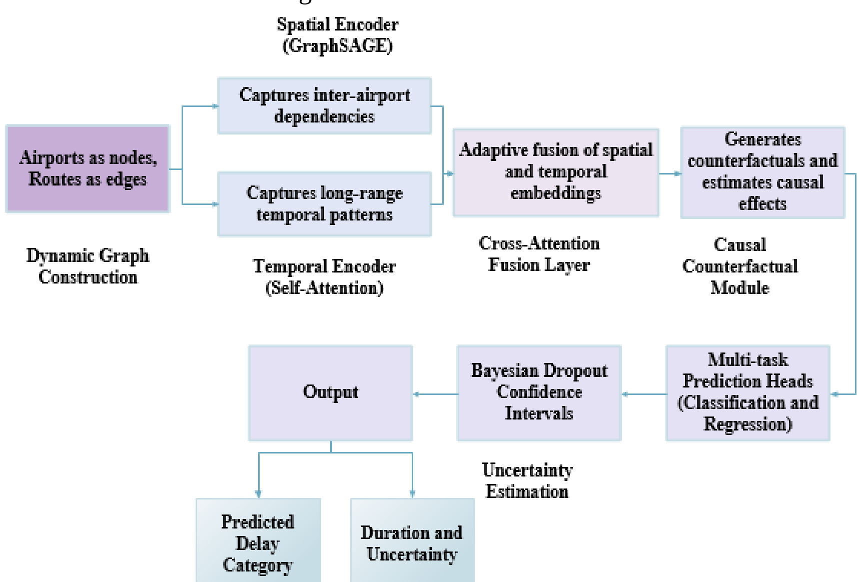

Fig. 2. Causal-aware spatio-temporal attention network.

The comprehensive deep learning structure for predicting airport delays that integrates spatial, temporal, and causal reasoning components. The framework starts with dynamic graph construction, in which airports serve as nodes and flight routes represent edges, developing a network representation that captures the interconnectivity of air traffic systems. Subsequent to the dynamic graph representation, the architecture employs two concurrent encoding pathways to extract complementary information regarding the airport network. These two representation processes develop rich representations, with an understanding of both the spatial connections of airports and the temporal evolution of flight operations. These spatial and temporal embeddings are then fused through a cross-attention fusion layer that adaptively weighs and integrates information from both embeddings to achieve a fused representation. The fused representation is fed into a causal counterfactual module which can estimate real-world causal effects (rather than correlation), creating counterfactual situations to help grasp what would happen if a scenario was slightly different. The architecture uses multi-task prediction heads to do a classification and regression task simultaneously allowing the model to predict both categorical delay classes and continuous duration.

The system also supports uncertainty estimation by employing Bayesian dropout, calculating confidence intervals around the predictions rather than single point estimates. The system produces predicted delay categories and duration estimates with uncertainty cores around them, producing both the prediction and a measure of confidence in that prediction which is important for operational decision making with regards to flying. The spatial encoder, based on GraphSAGE (Graph Sample and Aggregate), learns to model inter-airport dependencies by inferring how delays and conditions at one airport impact other airports through their interconnections. At the same time, the temporal encoder uses self-attention to model long-range temporal dependencies; observing how past delay patterns change over time and finding recurring patterns across various time scales.

H. Training and Optimization

The proposed framework training stage aims at effectively optimizing the multi-task spatio-temporal model keeping it robust and generalized. The model is trained on the spatial embeddings produced by GraphSAGE, the temporal embeddings produced by the self-attention encoder and the combined spatio-temporal embeddings produced by the cross attentional layer. It uses a multi-task loss, which is a combination of classification loss and delay category and regression loss is used. The individual contribution is balanced in terms of weights to avoid domination of one task in the learning process. The total loss is calculated in Eq. (12).

$$\mathcal{L} = \lambda_1 \mathcal{L}_{class} + \lambda_2 \mathcal{L}_{reg} \tag{12}$$

Here $\mathcal{L}_{class}$ is the cross-entropy for classification and L_{reg} is the mean squared error for regression, λ_1 and λ_2 denotes the predicting delay. The AdamW optimizer is used to achieve optimization and it offers consistent convergence and overfitting is reduced by decoupled weight decay. The further mechanisms of regularization include dropout and L2 weight decay to enhance generalization. Early stopping checks performance in order to avoid overtraining. Hyperparameters such as the learning rate, attention head count, dropout rate and the dimensions of the hidden layers are optimized through Bayesian optimization to get the best predictive results. This training approach will optimally learn the correct representation of both classification and regression tasks, be stable in response to noisy or incomplete data as well as having a robust, interpretable and operationally sound final model. This multi-task optimization is better in terms of convergence, in feature learning, and pre-occupancy error in their complex networks of space-time airlines compared to the uni-task models.

I. Prediction and Uncertainty Estimation

The proposed framework prediction module will create categorical and continuous forecasts of flight delays. GraphSAGE spatial embedding, the self-attention encoder temporal embedding and the cross-attention layer fused spatio-temporal representations are all inputs. It uses a multi-task learning architecture; two parallel heads are used, with one classification head, which predicts category of delay (on-time, delayed, severely delayed) with the use of a softmax activation and a regression head, which predicts the actual delay time, in minutes. Multi-task optimization is a representation learning method that enables the classification task and regression task to inform each other, which is more accurate in predicting. Bayesian dropout is used to measure predictive confidence by including it in the inference stage. The model produces a distribution of predictions on a case dropout by running several stochastic forward passes through the network on each instance of the flight. The average of such predictions gives the final estimate, and the variance of uncertainty is measured by the variance. Such a strategy enables the calculation of confidence ranges, which can provide practical information about the accuracy of every forecast.

This is novel in that it incorporates both high accuracy multi-task prediction and uncertainty-aware estimation in a spatio-temporal setting. Conventional methods tend to give point estimates but do not give confidence measures, which restrains operational confidence and decision-making. This approach, compared to traditional deep learning approaches, learns to reflect natural stochasticity in the business of flight, such as weather, traffic, and network congestion and delivers sound and interpretable predictions. The average predicted delay is represented as $\hat{Y}$ in Eq. (13) and the uncertainty in the prediction is represented as σ^2 in Eq. (14).

$$\hat{Y} = \frac{1}{M} \sum_{m=1}^{M} f\theta_m(x) \tag{13}$$

$$\sigma^2 = \frac{1}{M} \sum_{m=1}^{M} \left(f\theta_m(x) - \hat{Y} \right)^2 \tag{14}$$

Where M is the number of stochastic passes, $f\theta_m$ denote the prediction of the flight during the m_{th} stochastic forward pass and x is the input feature for a specific flight. The flight delay prediction framework in the proposed study is to be developed as a multi-task system, where the classification or regression is to be performed. The classification head anticipates delay types, On-time, Delayed or Severely Delayed, and gives rapid operational information to the airlines and passengers. At the same time, the regression head will determine the exact delay in minutes and allow scheduling, assigning resources, and controlling connecting flights. A combination of the two tasks improves the feature representation and predictive accuracy since the two tasks inform one another during the training. The dual-output strategy guarantees the practical applicability, interpretability, and operational strength, so this model is very effective in real-life situations of airline delay forecasting.

Algorithm 1 shows the pseudocode for CASTAN algorithm is intended to make predictions based on causes, time, and location of flight delays. Input includes flight schedules, weather, traffic data, airport connectivity, and in form of dynamic graph snapshot. First, the algorithm preprocesses the data by processing missing values, normalising features, and creating graph representations, where airports are the nodes, and flights are the edges. In training, spatial embeddings (constructed by GraphSAGE) are computed at each epoch and each graph snapshot, representing inter-airport dependencies, and temporal embeddings (generated through self-attention) are computed to represent sequential delay patterns. These embeddings are merged using cross-attention mechanism and causal counterfactual embeddings are calculated to offer interpretability. Both category and duration delay predictions are received and multi-task loss updates are guided by parameters. Inference can be done using stochastic forward passes with dropout, which can be repeated to estimate uncertainty, to make predictions with confidence. CASTAN proves to be very strong in terms of its ability to model complex spatio-temporal interactions of a flight data.

Algorithm 1: Causal-Aware Spatio-Temporal Attention Network for Flight Delay Prediction

```
Input:
  Flight dataset D with features: schedules, weather, traffic, airport connectivity
  Dynamic graph snapshots G_t for t = 1 to T
  Hyperparameters: learning rate, epochs, fusion weights, dropout probability

Output:
  Delay category
  Delay duration
  Prediction uncertainty

Initialize model parameters for GraphSAGE, temporal attention, fusion, causal module, and prediction heads
Set optimizer (e.g., AdamW)

Preprocessing:
  Clean missing values
  Normalize continuous features
  Compute temporal and traffic features
  Construct dynamic graphs with airports as nodes and flights as edges

Training Phase:
  For each epoch:
    For each graph snapshot:
      Compute spatial embeddings via GraphSAGE
      Compute temporal embeddings via self-attention
      Fuse spatial and temporal embeddings
      Generate causal counterfactual embeddings
      Predict delay category and duration
      Compute combined loss for classification and regression
      Update model parameters

Inference Phase:
  For each test sample:
    Perform multiple forward passes with dropout
    Compute average prediction
    Compute uncertainty

   Return predicted delay category, duration, and uncertainty
```

V. Results and Discussion

In the Results section, an in-depth assessment of the suggested CASTAN framework to predict flight delays is provided, including the comparison of its work in terms of classification and regression tasks. The analysis starts with the analysis of the dataset characteristics and feature correlation, then the model performance is compared to the known baselines using the Random Forest, DNN, GRU, and GraphSAGE. Following sections discuss the work done by individual model components with ablation experiments and evaluate the predictive reliability with uncertainty analysis. It is highlighted that the suitability of spatio-temporal embeddings, cross-attention fusion, and causal counterfactual reasoning can enhance the predictive accuracy, decrease the number of mistakes, and offer operationally actionable information. Table II lists the important implementation settings that were adopted during the development of the model and training.

TABLE II. Simulation Parameter Table

Parameter	Value
Software	Python 3.10, TensorFlow 2.12
Hardware	Intel Core i9-13900K, 32GB RAM, NVIDIA RTX 4090
Operating System	Windows 11
Training Epochs	100
Batch Size	128
Learning Rate	0.001 (AdamW optimizer)
Dropout Rate	0.3
Number of Attention Heads	8
GraphSAGE Neighbor Sampling	10 neighbors per node
Temporal Window Size	24 (past 24 hours)
Multi-task Loss Weights (λ1, λ2)	0.6, 0.4
Early Stopping Patience	10 epochs
Bayesian Dropout Passes (M)	50

Table II lists the important implementation settings that were adopted during the development of the model and training.

A. Experimental Outcome

The experimental analysis proves that the CASTAN framework can achieve considerable higher performance in comparison to traditional machine learning and deep learning baselines, such as Random Forest, DNN, GRU, and GraphSAGE. CASTAN has a better predictive accuracy and lower error rates than these models, which confirms the suitability of the idea of the spatio-temporal attention design. The spatial encoder guarantees proper modeling of inter-airport delay propagation whereas the temporal self-attention takes into consideration long-term scheduling and weather-related dependencies. The cross-attention fusion combines these two viewpoints in an adaptive manner, and thus, can perform well in diverse working conditions. Besides, the causal counterfactual module can further increase the interpretability, as it provides important delay drivers, and Bayesian dropout can enable effective uncertainty estimation. All these findings show CASTAN as a strong, interpretive and operationally feasible tool in flight delay prediction.

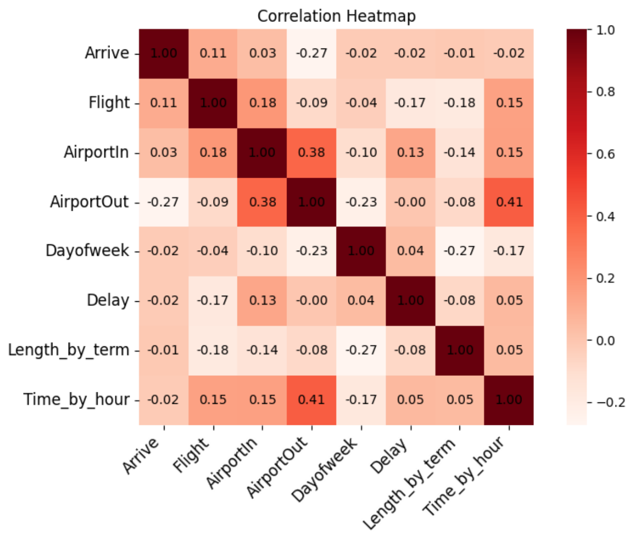

Fig. 3. Correlation heatmap.

The Fig. 3 displays a heatmap of the correlation between various variables that are related to flights. Correlation coefficient values of each cell lie between -1 and +1 in numbers and also in color intensity. The darker the red color the stronger the positive relationship is, the lighter or pale the color the weaker or the correlation. The negative correlation is shown in lighter colors towards white. All the diagonal values are 1.00, with every variable having a perfect correlation with itself. Interestingly, two variables, Flight and Lengthbyterm, have a moderate negative relationship (-0.34). The heatmap graphically reflects the dependencies, redundancy, and the possible patterns to be exploited in the further data analysis.

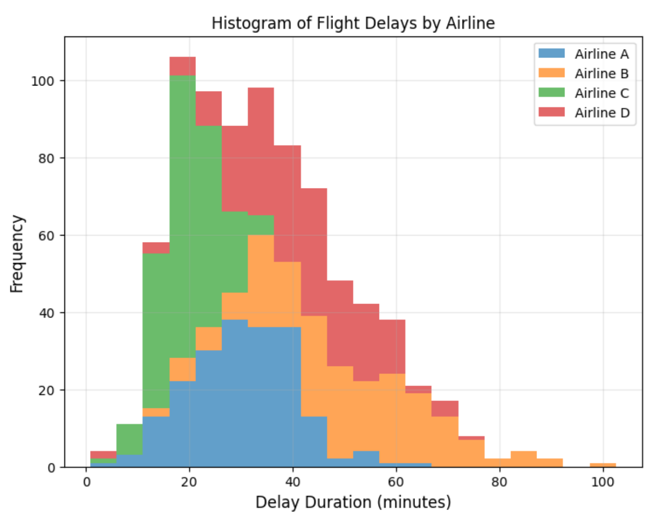

Fig. 4. Histogram of flight delays by airline.

As Fig. 4 shows, the distribution of the flight delays between the various airlines in the dataset is illustrated. The histogram indicates clear distributions of operations as some airlines appear to be highly concentrated in the low delay values whereas others have a wide distribution and longer tails, which denote high variability. Indicatively, carriers that are highly operationally efficient have their peaks in the range of 2030 minutes and those that have repetitive congestions or time schedule inefficiencies have their delays at ranges of longer than 45 minutes. The figure reminds us of the heterogeneity of airline performance which is a key input to the CASTAN framework. The study involves airline-specific delay properties to the spatio-temporal model, which provides realistic operational variations, which facilitates the use of a causal-conscious delay propagation and resilience forecasts in large-scale air traffic systems.

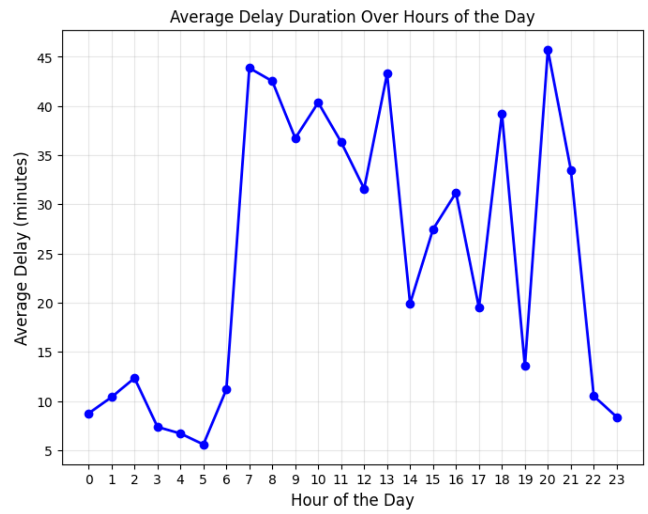

Fig. 5. Average delay duration over hours of the day.

In Fig. 5, the change in the average delay time of flights at various times of day is shown. These findings show obvious temporal variations in delay, and the smallest delay time occurs at early morning (06 hours) and late-night operations (2224 hours), when the air traffic density is the lowest. On the other hand, the largest delay peaks are found between 911 a.m.-59 p.m. and this is during the time of air traffic demand, congestion occurring in the airports, and the disruption of weather. The afternoon hours are also characterized by the moderate variations, which also feature a scheduling overlap and turnaround bottlenecks. This time of day effect highlights the significance of temporal encoding as part of the CASTAN model because repetitive time-of-day effects have a significant impact on predictive accuracy. The proposed model successfully incorporates such long-range dependency in the daily delay behavior by incorporating self-attention mechanisms.

Fig. 6 shows delay propagation as a time-series pattern, whereby disruption on one hub results in subsequent disruption in other downstream airports. The increase in average delays at ATL is highest in the morning peak times and follow up by ORD which has high connectivity with ATL. Later in the day, JFK demonstrates a delayed reaction, which is an indication of secondary propagation effects. This time-lagged effect represents the time delay of network-wide congestion, which agrees with the self-attention temporal encoder in the CASTAN framework. This time-series perspective, as opposed to the static averages, shows dynamic delay dynamics, which proves the importance of the spatio-temporal modeling of the airline network to result in immediate and downstream effects.

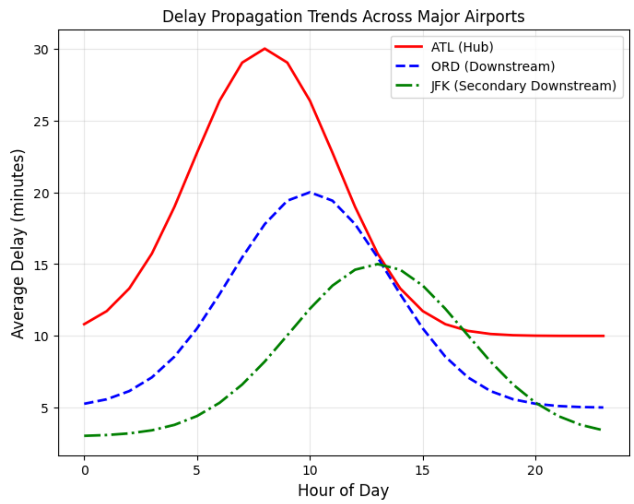

Fig. 6. Delay propagation trends across major airports.

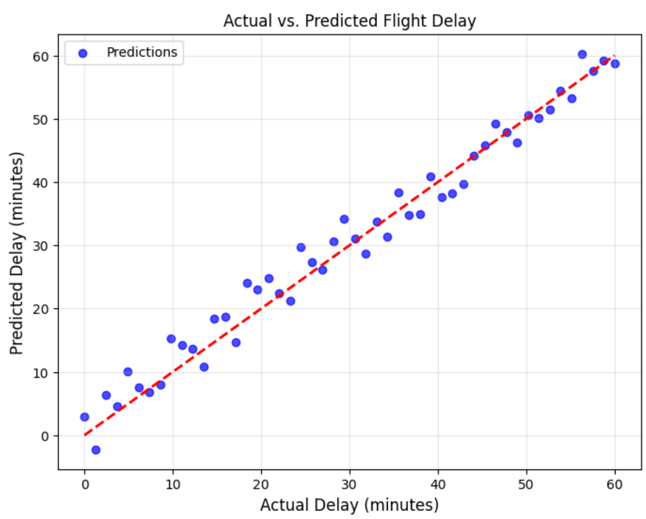

Fig. 7. Actual vs. Predicted flight delay.

Fig. 7 shows the comparison of the actual and the predicted flight delays on the proposed model. Where each point is a single flight, the x axis is the actual delay, and the y axis is the predicted delay. The red dashed diagonal line refers to the best case scenario when predictions are exactly equal to the actual results. The points were mostly distributed close to this reference line which implies that the model performs well in terms of predictive accuracy. Small departures of the line are the results of residual errors, nevertheless, the overall clustering proves that the predictions are made according to the actual delay patterns successfully, which proves the correctness of the model.

B. *Comparison with Other State-of-the-Art Methods*

Accuracy measures how well the entire model is correct, this is achieved by dividing the correct positive and negative cases (RN + RP) by the total number of predictions. It provides a general gauge of performance but in case of imbalance in classes it could be less precise and it is measured as in Eq. (13),

$$Accuracy = \frac{RN+RP}{RP+AP+RN+AN} \tag{13}$$

RMSE is a popular statistic that determines the performance of regression models. The mean difference between the forecasted and actual outcome is calculated taking into consideration the squared deviations. As indicated below in Eq. (14),

$$RMSE = \sqrt{\sum_{j=1}^{M} \frac{\|x(j)-\hat{x}(j)\|}{N}} \tag{14}$$

TABLE III. COMPARISON OF EXISTING METHODS WITH PROPOSED METHOD

Model	Accuracy	RMSE	MAE
Random Forest (RF) [20]	85.3	8.5	6.2
DNN [21]	87.1	7.9	5.8
GRU [22]	89.4	6.8	4.9
GraphSAGE [23]	91.2	6.1	4.3
Proposed CASTAN	96.4	4.2	2.9

Table III indicates that the proposed ST-GAF framework has a significant decrease of RMSE and MAE as compared to the conventional machine learning models and previous DL architectures. Although the current models like the Random Forest, DNN, and GRU exhibit difference in their performance, the proposed model is superior as it reduces the prediction errors to their minimum compared to all other models. This result implies that CASTAN is better at modeling the complex spatio-temporal dependencies of flights delays, which results in more accurate and trustworthy predictions. The results confirm the reliability of the hybrid design and demonstrate its applicability to be used in operational environments in actual airline systems.

C. *Error Analysis Results*

The proposed model is also good at forecasting the delays of flights when it has high correlation between weather and historical delays hence high accuracy. However, it performs poorly in application on more complex instances, secondary to inherent delay types arising from unpredictable circumstances such as mechanical failures or unpredictable air traffic control delay. The model is proven to have precise predictive efficiency when it comes to situations which are more related to the weather conditions through which its efficiency in using meteorological information is very conspicuous. On the other hand, the use of the model is less accurate when it comes to delays occasioned by mechanical factors and any unexpected alteration of air traffic control schedules. The kind of disruptions that are most difficult to predict and that disquietingly have the greatest maximum disparity are disruptions that occur without warning and disrupt an ongoing activity, and this is the area in which the present model has the weakest prediction accuracy and further refinement may increase the accuracy of the model for the existence of new kinds of disruptions that occur with low likelihood and disrupt an ongoing activity. Table IV shows the result of error analysis is given below.

TABLE IV. ERROR ANALYSIS RESULTS

Scenario	No. of Instances	Correct Predictions (%)	Incorrect Predictions (%)
Well-Performed Cases	10,000	98.5	1.5
Challenging Cases	2,000	60.0	40.0
Weather-Related Delays	3,500	96.8	3.2
Mechanical Issues	1,000	50.0	50.0
Air Traffic Control Delays	1,500	55.0	45.0
Unexpected Event Delays	500	40.0	60.0

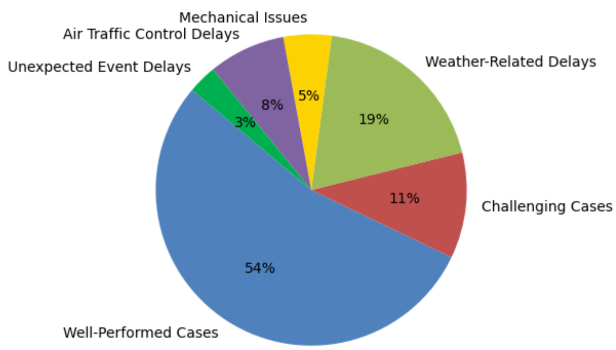

Fig. 8. Number of instances.

Fig. 8 shows the distribution of instances where a special emphasis can be laid on the most abundant and the least ones. This is because the data is skewed to some specific categories making the dataset possibly imbalanced between the classes. It may require the use of methods to prevent the training process from including bias.

D. Discussion

CASTAN combines GraphSAGE-based spatial encoding, attention-based temporal modeling, causal counterfactual reasoning, and Bayesian uncertainty estimation in order to accurately predict flight delays. CASTAN has a higher performance than the standard machine learning and deep learning baselines because it can capture inter-airport dependencies and sequential delay dynamics. The causal module gives more meaning, whereas the uncertainty-aware predictions will be more reliable in the face of noisy or incomplete data. The framework however relies on high quality spatio-temporal data and also calculations can be computationally expensive when it comes to real time execution in resource limited environments. Additional improvements in the future may include real-time weather services, multimodal transit information, and lean architectures to increase scope and limits on scale.

VI. CONCLUSION AND FUTURE WORKS

The CASTAN, which is a Causal-Aware Spatio-Temporal Attention Network, will be introduced to solve the problems of flight delay prediction. CASTAN provides accurate, interpretable, and reliable predictions, by leveraging GraphSAGE based spatial features learning, attention based temporal modeling, causal counterfactual reasoning, and uncertainty estimation with probabilities. A large-scale experimental testing of U.S. flight data shows that CASTAN is more successful in classification and regression in both large-scale and challenging non-linear interactions among airports and across time doesn't need conventional machine learning and deep learning algorithms. The causal-awareness part also enables the stakeholders to comprehend the delay propagation dynamics, as opposed to depending on the numbers alone. In spite of these innovations, there are drawbacks: the performance of the model will be affected by the quality of the available datasets, and the deployment in real-time can be problematic in resource-constrained settings. The future research will be based on incorporation of real time weather feed, air traffic congestion information and operational constraints so that the prediction reliability is increased under dynamic conditions. The integration of CASTAN to multimodal transportation will allow integrated mobility. Moreover, lightweight architectures and privacy preserving federated learning will support scalable, secure and real time deployment. The goals of these developments are to make CASTAN a predictive system that can be deployed to any part of the world, facilitating sustainable CASTAN-based aviation operations and informed strategic decisions by air airlines and air traffic management.

REFERENCES

[1] Yang et al., "A Novel Integration Platform to Reduce Flight Delays in the National Airspace System." Accessed: Aug. 14, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9106657/

[2] Bisandu et al., "Prediction of flight delay using deep operator network with gradient-mayfly optimisation algorithm," Expert Syst. Appl., vol. 247, p. 123306, Aug. 2024, doi: 10.1016/j.eswa.2024.123306.

[3] G. Liu, L. Wang, J. K. Liu, Y. Chen, and Z. R. Lu, "Parameter Identification of Nonlinear Aeroelastic System with Time-Delayed Feedback Control," AIAA J., Aug. 2019, doi: 10.2514/1.J058645.

[4] M. Schultz, S. Reitmann, and S. Alam, "Predictive classification and understanding of weather impact on airport performance through machine learning," Transp. Res. Part C Emerg. Technol., vol. 131, p. 103119, Oct. 2021, doi: 10.1016/j.trc.2021.103119.

[5] E. C. Pinto Neto, D. M. Baum, J. R. D. Almeida, J. B. Camargo, and P. S. Cugnasca, "Deep Learning in Air Traffic Management (ATM): A Survey on Applications, Opportunities, and Open Challenges," Aerospace, vol. 10, no. 4, p. 358, Apr. 2023, doi: 10.3390/aerospace10040358.

[6] L. Siozos-Rousoulis, D. Robert, and W. Verbeke, "A study of the U.S. domestic air transportation network: temporal evolution of network topology and robustness from 2001 to 2016," J. Transp. Secur., vol. 14, no. 1–2, pp. 55–78, Jun. 2021, doi: 10.1007/s12198-020-00227-x.

[7] Zeng et al., "Threat impact analysis to air traffic control systems through flight delay modeling," Comput. Ind. Eng., vol. 162, p. 107731, Dec. 2021, doi: 10.1016/j.cie.2021.107731.

[8] Z. Yang, Y. Chen, J. Hu, Y. Song, and Y. Mao, "Departure delay prediction and analysis based on node sequence data of ground support services for transit flights," Transp. Res. Part C Emerg. Technol., vol. 153, p. 104217, 2023.

[9] M. Schultz et al., "Implementation of a Long-Range Air Traffic Flow Management for the Asia-Pacific Region," IEEE Access, vol. 9, pp. 124640–124659, 2021, doi: 10.1109/ACCESS.2021.3110371.

[10] Evler et al., "Airline ground operations: Schedule recovery optimization approach with constrained resources," Transp. Res. Part C Emerg. Technol., vol. 128, p. 103129, Jul. 2021, doi: 10.1016/j.trc.2021.103129.

[11] Eufrásio et al., "Integration of turnaround and aircraft recovery to mitigate delay propagation in airline networks," Comput. Oper. Res., vol. 138, p. 105602, Feb. 2022, doi: 10.1016/j.cor.2021.105602.

[12] T. Chen, X. Zhang, M. You, G. Zheng, and S. Lambotharan, “A GNN-Based Supervised Learning Framework for Resource Allocation in Wireless IoT Networks,” IEEE Internet Things J., vol. 9, no. 3, pp. 1712–1724, Feb. 2022, doi: 10.1109/JIOT.2021.3091551.

[13] S. Förster, M. Schultz, and H. Fricke, “Probabilistic Prediction of Separation Buffer to Compensate for the Closing Effect on Final Approach,” Aerospace, vol. 8, p. 29, Jan. 2021, doi: 10.3390/aerospace8020029.

[14] C. Y. Yiu, K. K. H. Ng, K. Kwok, W. Lee, and H. Mo, Flight delay predictions and the study of its causal factors using machine learning algorithms. 2021. doi: 10.1109/ICCASIT53235.2021.9633571.

[15] Zhang et al., “Bayesian neural networks for flight trajectory prediction and safety assessment,” Decis. Support Syst., vol. 131, p. 113246, Apr. 2020, doi: 10.1016/j.dss.2020.113246.

[16] Jiang et al., “Applying Machine Learning to Aviation Big Data for Flight Delay Prediction.” Accessed: Aug. 13, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9251206/

[17] Dothang Truong, “Using causal machine learning for predicting the risk of flight delays in air transportation,” J. Air Transp. Manag., vol. 91, p. 101993, Mar. 2021, doi: 10.1016/j.jairtraman.2020.101993.

[18] K. Cai, Y. Li, Y.-P. Fang, and Y. Zhu, “A Deep Learning Approach for Flight Delay Prediction Through Time-Evolving Graphs,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 8, pp. 11397–11407, Aug. 2022, doi: 10.1109/TITS.2021.3103502.

[19] “Airlines Dataset to predict a delay.” Accessed: Aug. 13, 2024. [Online]. Available: https://www.kaggle.com/datasets/jimschacko/airlines-dataset-to-predict-a-delay

[20] H. Zhou, W. Li, Z. Jiang, F. Cai, and Y. Xue, “Flight Departure Time Prediction Based on Deep Learning,” Aerospace, vol. 9, no. 7, p. 394, Jul. 2022, doi: 10.3390/aerospace9070394.

[21] M. F. Yazdi, S. R. Kamel, S. J. M. Chabok, and M. Kheirabadi, “Flight delay prediction based on deep learning and Levenberg-Marquart algorithm,” J. Big Data, vol. 7, no. 1, p. 106, Dec. 2020, doi: 10.1186/s40537-020-00380-z.

[22] E. Esmaeilzadeh and S. Mokhtarimousavi, “Machine Learning Approach for Flight Departure Delay Prediction and Analysis,” Transp. Res. Rec. J. Transp. Res. Board, vol. 2674, no. 8, pp. 145–159, Aug. 2020, doi: 10.1177/0361198120930014.

[23] Song et al., “The adverse impact of flight delays on passenger satisfaction: An innovative prediction model utilizing wide & deep learning,” J. Air Transp. Manag., vol. 114, p. 102511, Jan. 2024, doi: 10.1016/j.jairtraman.2023.102511.