International Journal on Advanced Science Engineering Information Technology

Vol.14 (2024) No. 6
ISSN: 2088-5334

Anchored Self-Supervised Dynamic Graph Representation Learning for Aviation Data as A Fast Economic Indicator

Wismu Sunarmodo [a], Bayu Distiawan Trisedya [a,*]

[a] *Faculty of Computer Science, University of Indonesia, Depok, Indonesia*
*Corresponding author: *b.distiawan@cs.ui.ac.id*

***Abstract*—The Fast Economic Growth Indicator, a newly developed metric leveraging big data, provides policymakers with timely insights crucial for assessing the economic impact of policies or events. Among various open-source datasets, aviation data stands out as a potential indicator of rapid economic growth, given its inherent graph structure with airports as nodes and flight connections as edges. However, global flight data, being dynamic and complex, poses challenges in analysis. To glean comprehensive insights, it's imperative to condense this graph data into representative vector values while preserving node relationships. In this study, we utilize the dynamic graph node embedding method to quantify the influence levels of airports relative to each other. Traditional node embedding methods often prioritize homophily over structural equivalence, challenging directly extracting influence levels. To address this limitation, we introduce anchored dynamic graph node embedding, employing a virtual node as a reference point in embedding space to enable direct calculation of influence levels. These influence metrics are then compared to the GDP of airport regions. Using USA domestic flight data from 1988 to 2021 as a case study, our methodology demonstrates promising results, boasting a 0.94 correlation coefficient with national GDP and a 0.9 correlation coefficient with state Gross State Product (GSP). This research aims to advance dynamic graph node embedding methods towards structural equivalence rather than homophily, enhancing applicability to tasks emphasizing node structure over neighborhood proximity. An example of the benefits of this research is its utility in addressing Influence Maximization Problems within dynamic graphs.**

***Keywords*— Dynamic graph node embedding; virtual node; centrality; fast economic indicator; aviation.**

Manuscript received 16 Jun. 2024; revised 9 Sep. 2024; accepted 17 Dec. 2024. Date of publication 31 Dec. 2024.

I. INTRODUCTION

In 2020, the global COVID-19 outbreak significantly impacted nations worldwide, prompting widespread travel restrictions to curb the virus's spread. However, these measures also led to economic slowdowns, plunging many countries into recession [1]. Such events underscore the need for economic indicators to swiftly capture and respond to emergent crises, natural disasters, disease outbreaks, or other disruptive phenomena. Traditional indicators, like Gross Domestic Product (GDP), often entail lengthy compilation times involving multiple data sources, limiting their utility for real-time policymaking [2].

Recognizing this gap, Haldane et al. [2] outlined criteria for a new economic indicator: high correlation with official government economic data, reliance on open data sources for reproducibility, and historical consistency for trend tracking and prediction. In recent years, researchers and government agencies have turned to big data analytics to develop near-real-time economic indicators, leveraging diverse datasets such as shipping data, population mobility data, sentiment analysis of news articles, and social media analysis [3], [4], [5]. Many of these studies highlight the correlation between economic growth and population movement.

Aviation data is a crucial source among these indicators, reflecting global trade and tourism patterns. The International Air Transport Association (IATA) reports that air transportation accounts for a substantial portion of global trade and travel [6]. While IATA's air connectivity index provides insights into the correlation between aviation and the economy, it primarily considers first-order proximity metrics like passenger numbers and flight frequencies, neglecting nuances like supply chain dynamics and transit flights.

Aviation data, comprising flight schedules between airports, naturally forms a graph structure, facilitating deeper insights through graph analysis techniques. Traditional methods like node degree and centrality metrics offer valuable insights but require manual feature extraction and struggle to capture dynamic changes over time. Representational learning, or node embedding, offers a promising alternative, aiming to capture graph centrality or influence over time. However, most node embedding methods prioritize tasks like link prediction [7], [8], [9] or node classification [10], [11], [12]. They often overlook the quantitative measurement of influence between nodes. Consequently, distances within the embedding space may not reflect the nodes' significance in the real world.

In our research, which focuses on assessing the influence level of nodes within a dynamic graph, it's essential that distances in the embedding space accurately capture node influence in real space. However, achieving this requires measuring distances relative to a reference point. We propose an anchored dynamic graph embedding approach for graph centrality analysis. By introducing a virtual anchor node with minimal weight connecting to every graph node, we establish a reference point in the embedding space, enabling the calculation of node centrality relative to this anchor. We ensure that the embedding space distance accurately reflects node centrality by leveraging positive and negative sampling techniques based on node occurrence in random walks. Finally, we validate our approach by comparing graph centrality metrics with Gross Domestic Product (GDP) and Gross State Product (GSP) for airport regions, offering insights into their economic contributions.

II. Materials and Method

A. Static Node Embedding

Static node embedding techniques aim to encode nodes into vector representations, commonly employing either random walk or graph neural network (GNN) approaches. Random walk-based methods, like NodeSig [13], node degree-based random walk [14], Hub-aware random walk [15], and Role-aware random walk [16], focus solely on the graph structure, neglecting node attributes. In contrast, GNN approaches, such as Graph Convolutional Networks (GCN) [17], GraphSAGE [18], Graph Attention Network (GAT) [19], and Graph Isomorphism Network (GIN) [20], incorporate node attributes for more comprehensive embeddings. GCN utilizes convolutional operations to propagate information between nodes, while GraphSAGE samples neighborhood nodes before feature aggregation. GAT employs attention mechanisms to weigh neighboring nodes' contributions, enhancing flexibility adaptively. GIN maximizes the discriminative power of node representation, making it suitable for classification tasks. Several methods are typically modifications of previously mentioned methods, such as [21], an extension of GraphSAGE, or [22], which is based on GAT.

B. Dynamic Node Embedding

In dynamic scenarios, graphs evolve over time by adding or removing nodes or edges. Consequently, relying solely on static node embedding methods leads to information loss. According to Kazemi [23], a task is inherently tied to the graph's representation of change. Typically, two central representations emerge: discrete-time dynamic graph (DTDG) and continuous-time dynamic graph (CTDG). In DTDG, dynamic graphs manifest as static snapshots per period. In contrast, CTDG portrays graph dynamics through changing events, such as adding a new edge at a specific timestamp (e.g., *edge_addition, (v1, v2, t0)*). While DTDG offers a more straightforward approach, it risks information loss when selecting periods that exceed the frequency of graph changes. This trade-off underscores the importance of selecting an appropriate representation that aligns with the underlying graph's dynamics and the embedding task's objectives.

Dynamic graph embedding typically incorporates static graph embedding as a spatial component or structure. This spatial component encodes nodes at a specific period (t). To capture the graph's dynamics, a temporal component is integrated with the spatial component in a specific manner. There are various strategies employed for node embedding in dynamic graphs, but not limited to encompassing modification of node attributes or identities, graph sequence prediction, dynamic parameter strategies, and embedding sequence prediction. In the modification of node attributes or identities approach, time parameters are integrated into node attributes or identities, distinguishing the same node at different times as distinct entities. This results in an aggregated graph of all periods condensed into a single static graph. Studies like [24] exemplify this method.

Graph sequence prediction strategies typically utilize autoencoder schemes. Here, the encoder processes input from multiple graph periods (e.g., $t\text{-}n$ to $t\text{-}1$) to generate an embedding vector, while the decoder predicts the graph for the subsequent period. Examples include [25], which utilizes deep autoencoder architectures, and [26], which aggregates node features with their neighbors and previous periods through a convolution mechanism. The dynamic parameter strategy involves adjusting spatial component parameters for each period. This can entail integrating these parameters into models like RNNs. For example, [27] incorporates CNN parameters as input to the LSTM model for each period, while [28] employs an isolation strategy and expansion on GNN parameters. The embedding sequence prediction strategy predicts a series of embeddings, where a static embedding method generates embedding vectors for each period. These embeddings are then input into a temporal structure. For instance, [29] utilizes GCN as a spatial component and LSTM as a temporal component, while [30] utilizes the sleep-attention mechanism in both spatial (GAT) and temporal (multi-head self-attention) components.

Each approach presents unique advantages and disadvantages depending on the dataset characteristics and application. The approaches mentioned are not mutually exclusive, which means several approaches can be used simultaneously. Furthermore, researchers have the flexibility to separate spatial and temporal components using alternative methods based on their specific requirements.

C. Node Centrality

Graph-level centrality measures the importance of nodes relative to others within a graph, with various methods available depending on the perspective of node importance. Degree centrality, for instance, evaluates nodes based on their number of connections, emphasizing nodes with higher connectivity. Closeness and betweenness centrality gauge a node's importance by its proximity to others in the graph.

Eigenvector and Katz centrality, on the other hand, consider a node's significance concerning its neighbors' importance.

Some studies are dedicated to gauging the centrality or influence of nodes within the embedding domain. One prominent application is Influence Maximization Problems, to identify influential nodes within a graph. For instance, [31] and [32] employ MLP on embedding outcomes, training to rank nodes based on influence. Additionally, [33] utilizes struct2vec-generated embeddings as features for each node, subsequently fed into a GNN. Moreover, [34] leverages an autoencoder to produce embedding vectors, followed by dimension reduction for distance-based identification of influential nodes. Meanwhile, in [35], DeepWalk was employed to generate node embeddings, followed by calculating the Euclidean distance between each node and every other node. The lower the average distance of a node to others, the higher its centrality. Generally, the approach to identifying influential nodes involves a two-stage model: the first stage entails obtaining the embedding vector, while the second involves determining the node's influence level. The latter stage employs various approaches, including a deterministic method that calculates distances and a heuristic approach using MLP with the training target to rank node influence levels.

D. *Centrality in Dynamic Graph*

The proposed method's main idea is to use the distance between nodes in the embedding space to measure influence. To connect this distance to influence, the embedding model needs to be trained with a parameter that defines how much influence a node has. While traditional metrics like betweenness, closeness, node degree, or eigenvector centrality can be helpful, they do not fully capture the complex influence dynamics in dynamic graphs like flight data, where influence depends on connections and how often flights occur. To tackle this, the method uses a random walk to simulate how often airports are visited in the flight network. However, a fixed random walk intensity would give similar results to other centrality methods. Instead, the method suggests a flexible random walk that adjusts the total number of flights, so more flights have a higher visitation rate.

Another challenge is that existing centrality methods often rely on fixed reference points or ground truth data for training, which may not work well in unsupervised settings. Virtual nodes are added to the graph as reference points. These virtual nodes act as origin points, with centrality increasing as you move away from them and decreasing as you get closer. However, adding virtual nodes can affect the random walk process by giving them too much influence, so their weight must be minimized to keep it accurate. Virtual nodes make it harder for graph representation methods, especially Graph Neural Networks (GNNs), to learn effectively since GNNs aggregate features from neighboring nodes. An attention-based GNN method, like Graph Attention Networks (GAT), is used to address this. This allows the importance of connections to and from virtual nodes to be adjusted dynamically, ensuring the learning process remains robust.

E. *Anchored Dynamic Graph Embedding*

The anchored dynamic graph embedding method builds upon the DySAT model with several key modifications. The training process depicted in Fig. 1 primarily involves two main components: generating positive and negative samples for the loss function calculation and the actual training process itself. The DySAT model makes three notable distinctions: the inclusion of virtual nodes, proportional random walk, and the context pairing model.

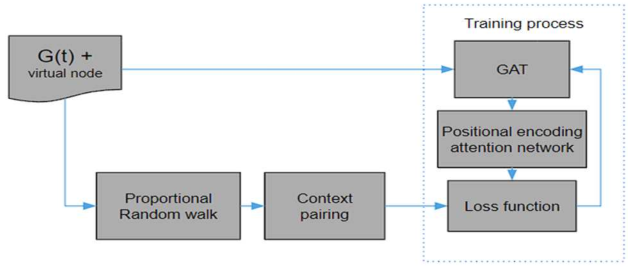

Fig. 1 Training process of self-supervised anchored dynamic graph embedding flow diagram

Virtual nodes are artificial nodes added to the original graph, acting as reference points in the embedding space to calculate each node's centrality. These virtual nodes are connected to every node in the graph with the minimum edge weight value. In the random walk process, the choice of the next target node depends on the probability of that node compared to its neighboring nodes from the source node. The probability of choosing a target node relies on two factors: the weight of the edge between the source node and the target node, and the total number of neighboring nodes connected to the source node. A higher edge weight from the source node to the target node increases the probability of selection. Additionally, if the source node has fewer neighboring nodes, the probability of selecting any single neighbor also increases.

Typically, nodes that are considered more significant tend to have many neighbors, each with high edge weights. Conversely, less significant nodes usually have fewer neighbors and lower total edge weights. Adding a virtual node with minimal edge weights connected to every node in the graph makes it more likely to be chosen during the random walk when the source node is less significant, as its probability becomes higher compared to other neighbors. On the other hand, if the source node is significant, the chance of selecting the virtual node decreases because its probability is lower than that of the more significant neighbors. Thus, during the embedding process, the virtual node with minimum edge weight will end up closer to the less significant nodes and further away from the significant ones.

In contrast to using a fixed walk length for all time steps, the proportional random walk approach calculates the walk length in proportion to the graph's total weight. Although there isn't a specific value for the divider of the total weight, a recommended walk length falls within the range of 20 to 50. Regarding context pairing, the choice of positive and negative context pairs is crucial for determining the model's objective function. Since the primary task of this research is to derive node centrality, node embedding should be based on structural equivalence rather than homophily. However, structural equivalence must also account for the edge weight. Positive and negative context pairs are sampled based on node occurrence rather than node degree to achieve this objective function. The proportional walk length ensures equitable node occurrence for each graph period, depending on the total weight of each graph. Thus, the loss function of this objective function is formulated as:

$$L = \sum_{t=1}^{T} \sum_{v \epsilon V_t} (\sum_{u \epsilon NP_t(v)} BCElogits(e_v^t . e_u^t) + w_n . \sum_{u \epsilon NN_t(v)} BCElogits(- e_v^t . e_u^t)) \quad (1)$$

The training process for embeddings is guided by the loss function used. In node embedding, the loss typically has two parts: positive loss and negative loss. Intuitively, the positive loss measures how close the target node's embedding is to nodes in the positive sample; the closer they are, the smaller the positive loss. Conversely, the negative loss decreases as the distance between the target node's embedding and nodes in the negative sample increases. The training process aims to minimize both positive and negative losses. Ultimately, the arrangement of node embeddings reflects a balance between positive and negative sampling.

The proposed method uses cosine similarity to measure the distance between node embeddings, which is then used in the BCE logits function. W_n represents the negative loss weight, usually set to a small value akin to the learning rate to facilitate better convergence. NP_t and NN_t denote the probability functions for positive and negative sampling of the target node v, respectively, defined as:

$$NP_t(v) = \frac{1}{abs(OC_{u \epsilon V}^t - OC_v^t) + 1} \quad (2)$$

$$NN_t(v) = abs(OC_{u \epsilon V}^t - OC_v^t) \quad (3)$$

Intuitively, nodes u in the vicinity of the target node v are more likely to be selected as positive samples. At the same time, those further away are more likely to be chosen as negative samples. Even though the virtual node boasts the highest node degree, its node occurrence is comparable to nodes with similar minimum edge weights during the random walk process. The Graph Attention Networks (GATs) model enables the integration of virtual nodes within the graph, as the training process dynamically adjusts the edges to or from nodes with varying node occurrences. This process of context pairing is shown in Fig. 2.

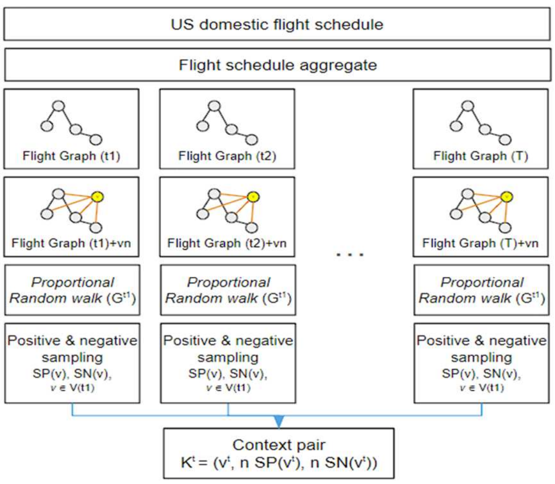

Fig. 2 In the self-supervised anchored dynamic graph embedding method, the context pairing process initiates with graph generation, adding virtual nodes, proportional random walk, and, ultimately, positive and negative sampling of the target node.

III. Results and Discussion

A. Dataset

The US domestic flight schedule data from 1988 to 2021 was sourced from Transtats, provided by the Bureau of Transportation Statistics (BTS) under the Department of Transportation (DOT). To model the dynamic nature of flight schedules, a discrete-time dynamic graph (DTDG) representation was employed, aggregating flight schedules over specified time intervals. Each year's data was then transformed into a graph, with flight frequency serving as the weight of the edges.

Additionally, domestic flight passenger data covering the period from 1990 to 2021, necessary for calculating the IATA air connectivity index, was obtained from BTS DOT. Moreover, the US Gross Domestic Product (GDP) and Gross State Product (GSP) data were retrieved from the US Bureau of Economic Analysis (BEA) historical data, released in 2022. The SAGDP1 index (State Annual Gross Domestic Product) summary 1, available from 1997 to 2021, was utilized for analysis.

B. Airport Centrality and Economic Growth

The initial experiment conducted involved training the proposed model using input data comprising aggregated aviation graph data from 1988 to 2021. For the graph centrality task, the node embeddings generated by the proposed methods are utilized to compute the cosine similarity relative to the virtual node. This process involves measuring the cosine similarity between each node embedding and the embedding of the virtual node, thereby quantifying the centrality of each node in relation to the reference point provided by the virtual node. Formulated as follow:

$$C_v^t = \frac{e_v^t . e_{vn}^t}{|e_v^t||e_{vn}^t|} \quad (4)$$

$$TC_{area}^t = \sum_{i=v}^{V} C_i^t \quad (5)$$

The centrality index (C_v^t) of node $v \epsilon V$ in graph G^t with e_{vn}^t as the embedding of the virtual node, is computed. TC_{area}^t represents the total centrality index at period t. This index is then compared to the US Gross Domestic Product (GDP), alongside other graph centrality methods. The calculation of the total centrality index applies to various methods including degree, betweenness, closeness, eigenvector, and Katz centrality. As a reference, the IATA (International Air Transport Association) air connectivity index is also utilized, calculated using the following formula [6]:

$$AC_v^t = w_v . nP \quad (6)$$

The IATA air connectivity index at airport v at time t is computed by multiplying the airport's weight by the number of passengers departing from that airport. Airport weight is contingent on the type of flights the airport serves; typically, international airports carry a higher weight than domestic ones. However, since all the data in this research pertains to domestic flights, all airport weights are assumed to have the same value, 1.

The proposed method's centrality index is then correlated with economic growth data (GDP), alongside several other centrality measurement methods and the IATA air connectivity index. The trend graph for each of these methods is displayed side by side in Fig. 3.

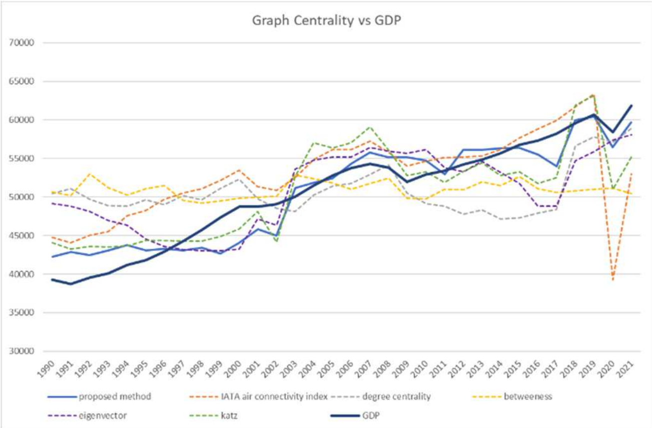

Fig. 3 Several methods linearly regressed total value of node centrality of aviation data and IATA air connectivity index compared to US GDP.

Visually, the US GDP graph reveals four significant declines (recessions): 1990-1991, 2001-2002 (related to the 9/11 attacks), 2007-2009 (Great Recession), and 2020 (COVID-19 pandemic). In contrast, the proposed method exhibits more declines, yet still captures the overall trend in GDP. This could be attributed to the fact that the data utilized is solely domestic flight data, whereas GDP is influenced by both domestic and international flights. Consequently, several domestic events, such as presidential elections (1992, 1996, 2000, 2004, 2008, 2012, and 2016), contribute to fluctuations in total centrality, resulting in discernible decreases during these periods.

The correlation indices between each method and GDP and GSP are summarized in Table 1. Three correlation testing scenarios were conducted: the average Pearson correlation index between the total area centrality of the country and GDP per year (1990-2022), the average Pearson correlation index between total centrality of state areas and GSP per year (1997-2021), and the Pearson correlation index between the concatenation of total state area centrality and GSP over time (1997-2021).

TABLE I
THE PEARSON CORRELATION INDEX OF EACH GRAPH CENTRALITY METHOD IS COMPARED TO THE US GROSS DOMESTIC PRODUCT (GDP) AND GROSS STATE PRODUCT (GSP).

Graph Centrality Methods	**Mean Annually Correlation Index to GDP (1990-2022)**	**Mean Annually Correlation Index to GSP (1997-2021)**	**Total Correlation Index to GSP (1997-2021)**
IATA air connectivity index	0.7249	0.8714	0.8552
Degree centrality	0.4760	0.8541	0.8395
Betweenness centrality	0.1432	0.4543	0.4517
Eigenvector centrality	0.7039	0.9211	0.9074
Katz centrality	0.8512	0.8526	0.8351
Proposed method	0.9413	0.9122	0.8972

The first experiment evaluates both spatial and temporal performance in line with GDP calculation, which aggregates state GSP within a country. By analyzing the average correlation between centrality indices and GSP, the model's stability to changes in centrality over time is clarified. This is particularly relevant because states usually have fewer airports than the national level, offering a clearer view of airport dynamics related to economic changes.

The proposed model outperforms other methods significantly in this experiment, demonstrating better spatiotemporal performance. The eigenvector centrality method performs best in the second and third tests, though only slightly better than the proposed method. However, its performance declines on a broader spatial scale. Overall, the proposed method shows a more balanced and stable performance in both spatial and temporal aspects, surpassing the IATA reference method.

This experiment also compares several representation learning methods on dynamic graphs to see how well they predict vertex centrality. Since other methods do not clearly convert the embedding vectors into centrality indices, we use proposed one, as predictors in a linear regression model. We use the IATA air connectivity index as the ground truth data because no data is available on the centrality index or economic contributions for each airport.

The Ridge linear regression method is employed as the training model, with input airport embedding vectors from each dynamic graph embedding method. The training target is the IATA air connectivity index for each airport. The prediction results are then compared with the IATA air connectivity index to calculate each method's Mean Squared Error (MSE).

TABLE II
MSE SCORE OF RIDGE REGRESSION OF DYNAMIC GRAPH EMBEDDING METHODS TO IATA AIR CONNECTIVITY INDEX.

Dynamic Graph Embedding Methods	**MSE to IATA air connectivity index**
DynGEM	0.12588
DynAE	**0.08689**
DynRNN	0.26178
DynAERNN	0.21320
CTGCN	0.55465
Proposed method	**0.07983**

The prediction error results for each method are summarized in Table 2. The proposed method exhibits the lowest error against the IATA air connectivity index, followed by DynAE. This indicates that the proposed method is well-suited for applications that characterize nodes based on their structural properties within a graph (structural equivalence). A comparison of the top 10 airport centrality rankings from several methods is presented in Table 3.

TABLE III
SAMPLE REFERENCE METHOD (IATA) AIRPORT CENTRALITY RANKING WITH THE PROPOSED METHOD AND DYNAE-BASED METHOD. THE HIGHLIGHTED CELLS ARE AIRPORTS THAT ARE RANKED IN THE TOP 10 OF THE IATA INDEX

1990			**2000**			**2010**			**2020**		
IATA	**Propsd**	**DynAE**	**IATA**	**Propsd**	**DynAE**	**IATA**	**Propsd**	**DynAE**	**IATA**	**Propsd**	**DynAE**
ORD	ORD	ORD	ATL	ORD	ORD	ATL	ORD	ATL	ATL	ORD	ATL
ATL	ATL	DFW	ORD	DFW	DFW	ORD	ATL	ORD	DFW	ATL	DEN
DFW	DFW	ATL	DFW	ATL	PHX	DEN	DFW	LAX	DEN	DFW	ORD
LAX	DEN	SFO	LAX	STL	LAX	DFW	DEN	DFW	ORD	DEN	DFW
SFO	STL	PHX	PHX	MSP	ATL	LAX	DTW	DEN	CLT	MSP	CLT
DEN	LAX	LAX	DEN	DTW	STL	PHX	MSP	PHX	LAX	LAX	LAX
PHX	PIT	DEN	LAS	PIT	DEN	LAS	LAX	SFO	LAS	IAH	PHX
LGA	PHX	DTW	STL	PHX	MSP	CLT	PHX	LAS	PHX	PHX	SEA
STL	CLT	STL	MSP	DEN	EWR	MCO	IAH	IAH	MCO	DTW	SFO
DTW	MSP	BOS	DTW	LAX	SFO	IAH	EWR	JFK	SEA	CLT	LAS

C. Experimenting with the model across different time aggregations.

The third experiment involves testing the model that has been trained and then using input graphs with different time aggregation types compared to the training phase. While annual graph data was used during training, the testing phase will involve flight data aggregated monthly. Suppose the previous tests focused more on examining the model's performance concerning spatial scale changes (airport-level, state-level, and country-level). In that case, this experiment evaluates whether the proposed model can adapt to temporal scale changes (from annual to monthly). During the training process, the proposed method utilized 35 time periods. Consequently, during testing using monthly aggregated data, node embeddings are generated for each of the 35 monthly periods without any overlap.

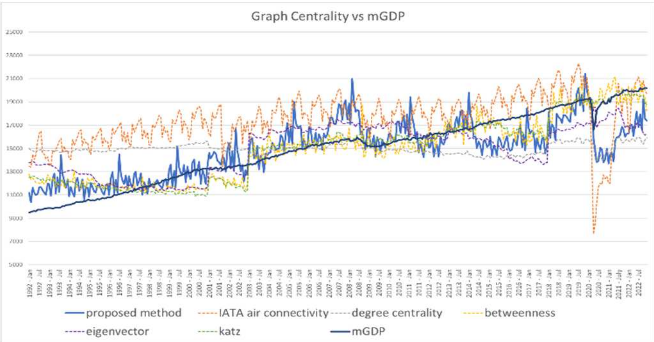

Fig. 4 The results of testing the proposed method using monthly aggregated flight schedule data are compared against other graph centrality methods and monthly GDP data for the United States from 1992 to 2022.

The results of the monthly total centrality calculations for the proposed method, comparative methods, reference method, and monthly GDP are displayed in Figure 4. Generally, monthly data yields more fluctuating total centrality graphs than annual data. Visually, the trend in the proposed method's graph generally aligns with the monthly GDP graph, depicting both increases and decreases, such as during the 2008 recession and the COVID-19 pandemic in 2020. However, there appears to be a recurring pattern in the proposed method's graph, especially noticeable in the repeated peaks every 35 months. Upon matching these peaks to the training data, they seem to coincide with the COVID-19 period in 2020, indicating that the temporal self-attention mechanism attempts to compensate for the downturn. This suggests that temporal self-attention and time position encoding may be less effective when applied to time series data, as patterns from the training data are carried over when testing with different time ranges or aggregations.

TABLE IV

COMPARISON OF CORRELATION INDICES BETWEEN TESTING THE PROPOSED METHOD, THE REFERENCE METHOD (IATA), AND GRAPH CENTRALITY METHODS WITH MONTHLY FLIGHT GRAPH INPUT DATA AGAINST MONTHLY GDP (S&P GLOBAL INDEX) FOR THE UNITED STATES.

Graph Centrality	**Mean Monthly Correlation Index to monthly GDP (1992-2022)**
IATA air connectivity index	0.5327
node centrality	0.1426
betweenness centrality	0.9181
eigenvector centrality	0.7215
Katz centrality	**0.9272**
Proposed method	0.7844

The correlation index results of centrality methods against monthly GDP are displayed in Table 4. The best correlation results are obtained with the Katz centrality method, followed by the betweenness centrality method, which even outperforms the results obtained with annual data. Generally, graph centrality methods yield better predictions than the IATA air connectivity index method. However, incorporating temporal components based on a combination of time position encoding and self-attention leads to poorer correlation index values than the previous testing. This is attributed to the compensatory patterns of the temporal component against the training data being carried over to the testing data, resulting in recurring patterns in the longer time range data.

IV. Conclusion

This study has delved into dynamic graph representation learning to ascertain node centrality within the graph. Employing virtual nodes and a loss function rooted in probability-based random walks with proportional steps, the resultant node embedding vectors can directly compute their centrality indices based on the degree of similarity with the artificial node embeddings. Comparative analyses with other methodologies, including reference standards (like the standard IATA), graph centrality techniques, and alternative dynamic graph learning approaches for GDP prediction, exhibited superior outcomes. Nevertheless, testing with varied time aggregations yielded suboptimal predictions due to recurrent patterns arising from the temporal components grounded in time position encoding and self-attention. However, overall, graph-centric centrality/connectivity analysis methods utilizing solely open-source data (such as flight schedules) generated relatively accurate predictions compared to standard references (IATA) relying on closed-source data (such as passenger counts on flights).

While the proposed method adeptly measures centrality indices, the potential utilization of node embeddings within the graph can extend to various applications, encompassing relationship prediction, node classification, and more. Despite the method's inclination towards structural equivalence rather than local similarity (homophily), it demonstrates robustness across diverse scenarios.

Including time position encoding and self-attention models within the temporal component markedly influenced testing across different time ranges or aggregations from the training data. To enhance outcomes, exploring tailored self-attention or transformer methods explicitly optimized for time series data may offer promising avenues.

This paper also tests the practical use of airport centrality measurements based on dynamic graph embedding to illustrate economic growth trends. The results show a strong correlation between an airport's centrality and the economic growth of its surrounding area. This approach allows us to assess the economic growth of regions connected by air transport in real-time. Stakeholders can use this data to guide policy decisions. Additionally, the impact of a policy can be reflected in the changes in airport centrality in those areas.

Furthermore, the dynamic graph embedding method can work with heterogeneous data. For example, it can combine air, land, and sea transport modes. It uses airports, terminals, stations, and ports as nodes, with travel relationships, passenger numbers, or travel frequency as edges and weights.

This way, we can gain a more comprehensive view of economic growth trends across different transport modes.

References

[1] M. Belitski, C. Guenther, A. S. Kritikos, and R. Thurik, "Economic effects of the COVID-19 pandemic on entrepreneurship and small businesses," *Small Business Economics*, vol. 58, no. 2, pp. 593–609, Feb. 2022, doi: 10.1007/s11187-021-00544-y.

[2] A. Haldane and S. Chowla, "Fast economic indicators," *Nature Reviews Physics*, vol. 3, no. 2. Springer Nature, pp. 68–69, Feb. 01, 2021. doi: 10.1038/s42254-020-0236-y.

[3] P. Aguilar, C. Ghirelli, M. Pacce, and A. Urtasun, "Can news help measure economic sentiment? An application in COVID-19 times," *Econ Lett*, vol. 199, p. 109730, Feb. 2021, doi:10.1016/j.econlet.2021.109730.

[4] G. Aoki, K. Ataka, T. Doi, and K. Tsubouchi, "Data-driven estimation of economic indicators with search big data in discontinuous situation," *The Journal of Finance and Data Science*, vol. 9, p. 100106, Nov. 2023, doi: 10.1016/j.jfds.2023.100106.

[5] S. Lehrer, T. Xie, and X. Zhang, "Social media sentiment, model uncertainty, and volatility forecasting," *Econ Model*, vol. 102, p. 105556, Sep. 2021, doi: 10.1016/j.econmod.2021.105556.

[6] IATA, "IATA 2022 Air Connectivity Report," 2022.

[7] Y. Zhang, Y. Xiong, D. Li, C. Shan, K. Ren, and Y. Zhu, "CoPE: Modeling Continuous Propagation and Evolution on Interaction Graph," in *International Conference on Information and Knowledge Management, Proceedings*, Association for Computing Machinery, Oct. 2021, pp. 2627–2636. doi: 10.1145/3459637.3482419.

[8] F. Aziz, L. T. Slater, L. Bravo-Merodio, A. Acharjee, and G. V. Gkoutos, "Link prediction in complex network using information flow," *Sci Rep*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-41476-9.

[9] P. Mei and Y. hong Zhao, "Dynamic network link prediction with node representation learning from graph convolutional networks," *Sci Rep*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-023-50977-6.

[10] M. Zou, Z. Gan, R. Cao, C. Guan, and S. Leng, "Similarity-navigated graph neural networks for node classification," *Inf Sci (N Y)*, vol. 633, pp. 41–69, Jul. 2023, doi: 10.1016/j.ins.2023.03.057.

[11] Y. Zhu, J. Wang, J. Zhang, and K. Zhang, "Node Embedding and Classification with Adaptive Structural Fingerprint," *Neurocomputing*, vol. 502, pp. 196–208, Sep. 2022, doi:10.1016/j.neucom.2022.05.073.

[12] R. Bhattacharya, N. K. Nagwani, and S. Tripathi, "A community detection model using node embedding approach and graph convolutional network with clustering technique," *Decision Analytics Journal*, vol. 9, p. 100362, Dec. 2023, doi:10.1016/j.dajour.2023.100362.

[13] A. Celikkanat, F. D. Malliaros, and A. N. Papadopoulos, "NodeSig: Binary Node Embeddings via Random Walk Diffusion," in *Proceedings of the 2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 68–75. doi: 10.1109/asonam55673.2022.10068621.

[14] S. N. Mohammed and S. Gündüç, "Degree-based random walk approach for graph embedding," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, no. 5, pp. 1868–1881, 2022, doi: 10.55730/1300-0632.3910.

[15] A. Tomčić, M. Savić, and M. Radovanović, "Hub-aware random walk graph embedding methods for classification," *Stat. Anal. Data Min.*, vol. 17, no. 2, Apr. 2024, doi: 10.1002/sam.11676.

[16] H. Zhang, G. Kou, Y. Peng, and B. Zhang, "Role-aware random walk for network embedding," *Inf Sci (N Y)*, vol. 652, p. 119765, Jan. 2024, doi: 10.1016/j.ins.2023.119765.

[17] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[18] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, in NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 1025–1035.

[19] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," *International Conference on Learning Representations*, 2018, [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ.

[20] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How Powerful are Graph Neural Networks?" in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=ryGs6iA5Km.

[21] W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher, and M. Portmann, "E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, IEEE Press, 2022, pp. 1–9. doi:10.1109/noms54207.2022.9789878.

[22] K. Yang, Y. Liu, Z. Zhao, X. Zhou, and P. Ding, "Graph attention network via node similarity for link prediction," *The European Physical Journal B: Condensed Matter and Complex Systems*, vol. 96, no. 3, pp. 1–10, Mar. 2023, doi: 10.1140/epjb/s10051-023-0.

[23] S. M. Kazemi *et al.*, "Representation Learning for Dynamic Graphs: A Survey," 2020. [Online]. Available: http://jmlr.org/papers/v21/19-447.html.

[24] X. Huang, J. Li, and Y. Yuan, "Link Prediction in Dynamic Social Networks Combining Entropy, Causality, and a Graph Convolutional Network Model," *Entropy*, vol. 26, no. 6, 2024, doi:10.3390/e26060477.

[25] X. Mo, J. Pang, and Z. Liu, "Deep autoencoder architecture with outliers for temporal attributed network embedding," *Expert Syst Appl*, vol. 240, p. 122596, Apr. 2024, doi: 10.1016/j.eswa.2023.122596.

[26] L. Liu, H. Zhao, and Z. Hu, "Graph dynamic autoencoder for fault detection," *Chem Eng Sci*, vol. 254, p. 117637, Jun. 2022, doi: 10.1016/J.CES.2022.117637.

[27] E. Yu, Y. Fu, J. Zhou, H. Sun, and D. Chen, "Predicting Critical Nodes in Temporal Networks by Dynamic Graph Convolutional Networks," *Applied Sciences (Switzerland)*, vol. 13, no. 12, Jun. 2023, doi:10.3390/app13127272.

[28] P. Zhang *et al.*, "Continual Learning on Dynamic Graphs via Parameter Isolation," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, in SIGIR '23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 601–611. doi:10.1145/3539618.3591652.

[29] J. Liu, C. Xu, C. Yin, W. Wu, and Y. Song, "K-Core Based Temporal Graph Convolutional Network for Dynamic Graphs," *IEEE Trans Knowl Data Eng*, vol. 34, no. 8, pp. 3841–3853, 2022, doi:10.1109/tkde.2020.3033829.

[30] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, "DYSAT: Deep neural representation learning on dynamic graphs via self-attention networks," in *WSDM 2020 - Proceedings of the 13th International Conference on Web Search and Data Mining*, Association for Computing Machinery, Inc, Jan. 2020, pp. 519–527. doi:10.1145/3336191.3371845.

[31] R. Bhattacharya, N. Nagwani, and S. Tripathi, "Detecting influential nodes with topological structure via Graph Neural Network approach in social networks," *International Journal of Information Technology*, vol. 15, Jul. 2023, doi: 10.1007/s41870-023-01271-1.

[32] M. R. F. Mendonça, A. M. S. Barreto, and A. Ziviani, "Approximating Network Centrality Measures Using Node Embedding and Machine Learning," *IEEE Trans Netw Sci Eng*, vol. 8, no. 1, pp. 220–230, 2021, doi: 10.1109/tnse.2020.3035352.

[33] S. Kumar, A. Mallik, A. Khetarpal, and B. S. Panda, "Influence maximization in social networks using graph embedding and graph neural network," *Inf Sci (N Y)*, vol. 607, pp. 1617–1636, Aug. 2022, doi: 10.1016/j.ins.2022.06.075.

[34] X. H. Yang *et al.*, "Identifying influential spreaders in complex networks based on network embedding and node local centrality," *Physica A: Statistical Mechanics and its Applications*, vol. 573, p. 125971, Jul. 2021, doi: 10.1016/j.physa.2021.125971.

[35] Q. Hu, J. Jiang, H. Xu, and M. Kassim, "IMNE: Maximizing influence through deep learning-based node embedding in social network," *Swarm Evol Comput*, vol. 88, p. 101609, Jul. 2024, doi:10.1016/j.swevo.2024.101609.