STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **14**, August 2025, pp 718–735.
Published online in International Academic Press (www.IAPress.org)

Application of Rainbow Vertex Antimagic Coloring in Multi-Step Time Series Forecasting for Efficient Railway Passenger Load Management

Dafik[1,2,*], Elsa Yuli Kurniawati[1], Ika Hesti Agustin[1,2], Arika Indah Kristiana[1,3], Robiatul Adawiyah[1,3], M. Venkatachalam[4]

[1]*PUI-PT Combinatorics and Graph, CGANT Research Group, University of Jember,Jember, Indonesia*
[2]*Department of Mathematics, Faculty of Mathematics and Natural Sciences, University of Jember, Jember, Indonesia*
[3]*Department of Mathematics Education, Faculty of Teacher Training and Education, University of Jember, Jember, Indonesia*
[4]*PG and Research Department of Mathematics,Kongunadu Arts and Science College, Coimbatore-641 029, Tamil Nadu, India*

Abstract Let G be a simple graph and connected. If there is a bijection function $f : E(G) \to \{1, 2, \cdots, |E(G)|\}$ and the rainbow vertex antimagic coloring is under the condition all internal vertices of a path $x - y$ for any two vertices x and y have different weights $w(x)$, where $w(x) = \Sigma_{xx' \in E(G)} f(xx')$. The least number of colors used among all rainbow colorings produced by rainbow vertex antimagic labelings of a graph G is the rainbow vertex antimagic connection number, $rvac(G)$. Our goal in this study is to prove some theorems related to $rvac(G)$. Furthermore, we apply RVAC as an administrative operator that controls passenger load anomalies at stations. This control uses spatio temporal multivariate time series Graph Neural Network (GNN) forecasting. Based on the results, we found that the metric evaluation of our GNN outperformed other models such as HA, ARIMA, SVR, GCN and GRU.

Keywords Rainbow Vertex Antimagic Coloring, Time Series Forecasting, Spatial Temporal Graph Neural Networks, Railway Station Passengers Load.

AMS 2010 subject classifications 68T07, 68R10, 05C90, 05C78

DOI: 10.19139/soic-2310-5070-2214

1. Introduction

Let $G = (V, E)$ be a connected, undirected and simple graph with the set of vertices $V(G)$ and the set of edges $E(G)$. A coloring on the edges of G with $c : E(G) \longrightarrow \{1, 2, \ldots, k\}$, $k \in N$, where the color of nearby edges can be the same. If no two edges of a $u - v$ path in G have the same color, the path is called a rainbow path. If every two vertices u and v of $V(G)$ have a rainbow $u - v$ path, then G is a rainbow connected graph according to c. If k colors can be used, then c is a rainbow k coloring. In this case the coloring c is called a rainbow coloring of G. The rainbow connection number $rc(G)$ of G is the lowest k for which a rainbow k-coloring of the edges of G exists [1]. A minimal rainbow coloring of G is defined as a rainbow coloring of G using $rc(G)$ colors.

Krivelevich and Yuster proposed a concept of the rainbow vertex-connection. Some results about rainbow vertex coloring of some graphs we can see in [2, 3, 4, 5]. If a path connecting any two vertices in a vertex-colored network has different colors for each internal vertex, then the graph is said to be rainbow vertex-connected. A rainbow vertex-connected path is the name given to such a path $rvc(G)$ is the rainbow vertex-connection count of a connected graph G. [3]. The minimum number of colors required to make G a vertex-connected rainbow is $rvc(G)$. They have provided the lower bound of rvc(G),

*Correspondence to: Dafik (Email: d.dafik@unej.ac.id). Department of Mathematics, Faculty of Mathematics and Natural Sciences, University of Jember, Jember, Indonesia.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

$$rvc(G) \geq diam(G) - 1.$$

Dafik *et al* [7] presented rainbow antimagic coloring, a new idea. Let G be a graph with vertices. A bijection f from the vertex set $V(G)$ to the set $1, 2, ..., |V(G)|$ is a labeling of the graph G. If for any two edges uv and $u'v'$ in the path between x and y we have $w(uv) \neq w(u'v')$, where $w(uv) = f(u) + f(v)$ and $x, y \in V(G)$, then this bijection f is called a rainbow antimagic vertex labeling. If a graph G has such a rainbow antimagic labeling, it is said to have a rainbow antimagic connection. Each rainbow antimagic labeling causes G to be colored in a rainbow fashion, with a color $w_f(uv)$ assigned to each edge uv. The smallest number of colors needed among all rainbow colorings generated by the rainbow antimagic labeling of G is the rainbow antimagic connection number of G, represented by $rac(G)$. Some researchers have developed the concept of rainbow antimagic coloring of graphs [6, 9].

In 2021, Marsidi et al [10] came up with the innovative idea known as rainbow vertex antimagic coloring. If all internal vertices of a path $x - y$ have different weights $w(x)$ for any two vertices x and y, where $w(x) = \Sigma_{xx' \in E(G)} f(xx')$, then there exist a bijection function $f : E(G) \rightarrow \{1, 2, \cdots, |E(G)|\}$ and the rainbow vertex antimagic coloring. The rainbow vertex antimagic connection count of a graph G is the least number of colors used in all rainbow colorings created by rainbow vertex antimagic labelings of G. This number is represented as $rvac(G)$. Our goal in this study is to prove some theorems related to $rvac(G)$. Several findings about the rainbow antimagic vertex coloring of some of the graphs in [5, 8, 10, 11].

Lemma 1.1
Let G be any connected graph. Then, $rvac(G) \geq rvc(G)$. [10]

A tadpole graph is a graph obtained by appending a path to a cycle. The tadpole graph has a geometric structure that is relevant to the placement of stations in East Java. In a tadpole graph, there is one vertex that acts as the "head" or "center" representing the main terminal, which is then connected to other vertices representing the surrounding stations. This structure can reflect a transport network centered on a large city with branches to other cities or regions around it. The tadpole graph is able to represent the concept of space required in passenger density management. By having a "center" node and branch nodes representing surrounding stations, tadpole graphs can show the spatial relationship between the stations. In addition, based on Rainbow Vertex Antimagic Coloring research, no one has examined tadpole graphs. Therefore, in addition to the use of rainbow vertex antimagic coloring, the innovation of this paper consists in some theorems about rainbow vertex antimagic coloring $rvac(G)$. The illustration of tadpole graph in Figure 1.

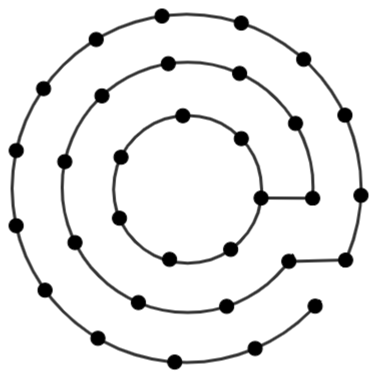

Figure 1. Tadpole graph.

Graph Neural Network (GNN) is an approach in the field of machine learning specifically designed to analyse data in the form of graphs. Graphs are complex structural representations, where entities (nodes) are connected by relations (edges). The non-Euclidean distance nature of graph data makes conventional approaches such as feedforward neural networks less effective. However, GNN utilises the spatial connection structure between nodes in the graph to understand its contextual information. As such, GNNs are able to better extract patterns and important information from graph data, making them suitable for solving railway station passenger crowding problems. The Spatial Temporal Graph Neural Network (STGNN) is a type of GNN. Some GNN related research results are [12, 13, 15, 16, 17, 18, 19, 20].

In addition, railway station passenger load management involves planning and executing strategies to efficiently handle passenger flow, ensuring smooth station operations, on-time train departures, and overall passenger satisfaction. One effective approach to tackling this challenge is graph coloring, specifically Rainbow Vertex Antimagic Coloring (RVAC). Unlike traditional graph coloring methods, RVAC uniquely models station connectivity while incorporating spatial and temporal variations in passenger flow. Its ability to assign distinct values to connected stations allows for a more structured representation of railway networks, making it particularly suitable for multi-step time series forecasting in passenger load management. By leveraging this approach, RVAC enhances the accuracy of passenger movement predictions, supporting more effective forecasting and optimized resource allocation. In this study, we apply RVAC in combination with Spatial-Temporal Graph Neural Networks (STGNN) to develop a robust predictive framework for railway passenger load management.

2. Method

This research adopts an approach that combines analytical and experimental methods. Our analytical approach involves deductive proof inspired by the principles of mathematical science. With this approach, we systematically validate the underlying theories of the railway passenger density problem in East Java. On the other hand, our experimental approach leads to the application of the results of such deductive proofs in a real context. Through experiments and empirical testing, we apply these theories to solve the problem of admin demand management for railway passenger crowding control. By combining these two approaches, we aim to provide a more comprehensive understanding and a more effective solution to the problems we study. The first process we do is embedding the Graph Neural Network feature. We use the Algorithm 1 of Single Layer Graph Neural Network (GNN) to perform the embedding process [14]. Each node in the graph has a feature that needs to be extracted using the GNN embedding process. The GNN embedding process involves two stages, namely Neural Message Passing (NMP) and Neural Message Aggregation (NMA). An illustration of this GNN embedding process can be seen in Figure 2.

Algorithm 1: Single Layer Graph Neural Network

Input : graph data $G(V, E)$, matrix adjacency A from graph G, matrix feature $H_{n\times m}$, and tolerance ϵ
Output: forecasting results

Initialize weights W, bias β, learning rate α.
for *error* $< \epsilon$ **do**
    Message passing $\mathbf{m_u}^l = MSG^l(h_u^{l-1})$.
    Aggregate the message $h_v^l = AGG^l\{m_u^{l-1}, u \in N(v)\}$.
    Determine the $error^l = \frac{||h_{v_i} - h_{v_j}||_2}{|E|}$.
    Update the learning weight $W^{l+1} = W_j^l + \alpha \times z_j \times e^l$
end
Save embedding results into a vector.
Open the embedded data.mat file.
To forecast, use time series forecasting.
Possess the greatest testing, training, and forecasting outcomes.

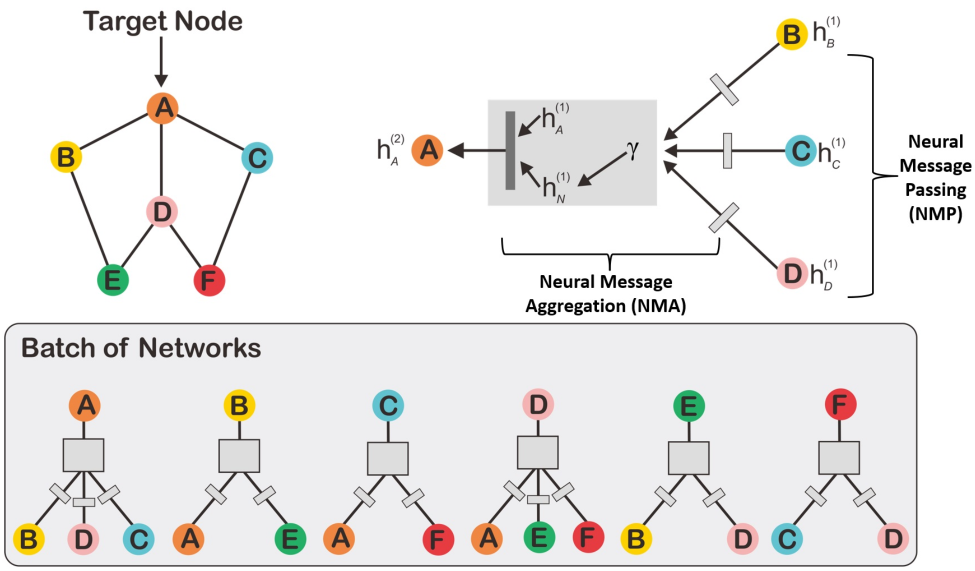

Figure 2. Illustration of Embedding Process in Graph Neural Network (GNN).

3. Main results

In this section we have the new theorem about the connection number of rainbow vertex antimagic coloring of some graph such as Cycle Graph (C_n) and Tadpole Graph $(T_{n,m})$.

Theorem 3.1
Let C_n be the cycle graph. For any integer number $n \geq 4$ and $n \equiv 0, 1, 2\ (mod\ 4)$, the connection number of rainbow vertex antimagic coloring of C_n is

$$rvac(C_n) = \begin{cases} \frac{n}{2} & , \text{for } n \equiv 2(mod\ 4) \\ \lceil \frac{n}{2} \rceil & , \text{for } n \equiv 1(mod\ 4) \end{cases}$$

$$\frac{n}{2} \leq rvac(C_n) \leq \frac{n}{2} + 1, for\ n \equiv 0(mod\ 4)$$

Proof. A cycle graph C_n contains the set of vertex $V(C_n) = \{x_i; 1 \leq i \leq n\}$ and the set of edges $E(C_n) = \{x_i x_{i+1}; 1 \leq i \leq n-1\} \cup \{x_1 x_{n-1}\}$. where $|V(C_n)| = n$ is the cardinality of the vertices and $|E(C_n)| = n$ is the cardinality of the edges. We find that the diameter of the cycle graph is $\lceil \frac{n}{2} \rceil$ to show the lower bound of the antimagic coloring of the rainbow vertex. Next we need to show the upper bound. We define antimagic labeling as a bijection f from $E(C_n)$ to the set $\{1, 2, ..., |E(C_n)|\}$ to demonstrate the upper bound. We present three scenarios to demonstrate the upper bound of the rainbow antimagic coloring of the cycle graph.

Case 1.For $n \equiv 2(mod\ 4)$.
Firstly, we show the bijective function of edge labels as follows:

$$f(x_i x_{i+1}) = \begin{cases} 2i+2 & , \text{for } i \equiv 1(mod\ 2) \text{ and } 1 \leq i \leq \frac{n}{2} - 1 \\ 2i+1 & , \text{for } i \equiv 0(mod\ 2) \text{ and } 1 \leq i \leq \frac{n}{2} - 1 \\ 2i-n+2 & , \text{for } i \equiv 1(mod\ 2) \text{ and } \frac{n}{2} + 1 \leq i \leq n-1 \\ 2i-n+1 & , \text{for } i \equiv 0(mod\ 2) \text{ and } \frac{n}{2} + 1 \leq i \leq n-1 \end{cases}$$

$$f(x_1 x_n) = 1 \quad f(x_{\frac{n}{2}} x_{\frac{n}{2}+1}) = 2$$

The vertex weight can be determined from the edge label function, such that we obtain:

$$w(x_i) = \begin{cases} n+1 & \text{, for } i = \frac{n}{2}, n \\ 4i+1 & \text{, for } 1 \leq i \leq \frac{n}{2} - 1 \\ 4i - 2n + 1 & \text{, for } \frac{n}{2} + 1 \leq i \leq n - 1 \end{cases}$$

We assumed that W_1 is the collection of vertex weights of C_n based on the vertex weights mentioned above and we have $W_{1,1} = \{n+1\}, W_{1,2} = \{5, 9, 13, \dots, 2n-3\}$, dan $W_{1,3} = \{5, 9, 13, \dots, 2n-3\}$. Furthermore, to show the vertex weight in $W_{1,1} \neq W_{1,2}$, we check if there is an element $W_{1,1}$ in $W_{1,2}$, given the assumption $W_{1,1} \subseteq W_{1,2}$, then

$$w(x_i) = w(x_i) \Longleftrightarrow n + 1 = 4i + 1 \Longleftrightarrow n = 4i \Longleftrightarrow i = \frac{n}{4}$$

Based on the assumption above, it is contradiction with the case because for this case we prove to $n \equiv 2(mod\ 4)$. So, we know the vertex weight in $W_{1,1} \neq W_{1,2}$. For $W_{1,2} = W_{1,3} = \{5, 9, 13, \dots, 2n-3\}$, we should find the cardinality using an arithmetic sequence $\{5, 9, 13, \dots, 2n-3, \}$.

$$U_s = a + (s-1)d \Longleftrightarrow 2n - 3 = 5 + (s-1)4 \Longleftrightarrow 2n - 3 = 4s + 1 \Longleftrightarrow s = \frac{n}{2} - 1$$

Given the vertex weights, we can compute the cardinality of W_1 as follows:

$$\begin{aligned} |W_1| &= |\{5, 9, 13, \dots, 2n-3, n+1\}| \\ |W_1| &= \frac{n}{2} - 1 + 1 \\ |W_1| &= \frac{n}{2} \end{aligned}$$

For $n \equiv 2(mod\ 4)$ we know that the rainbow vertex antimagic labeling of C_n has $\frac{n}{2}$ different weights. Each vertex $u, v \in V(C_n)$ has a color assigned to it, indicating that the vertex weights of each internal vertex u and v are different. The vertex weight makes it clear that they have a different weight.

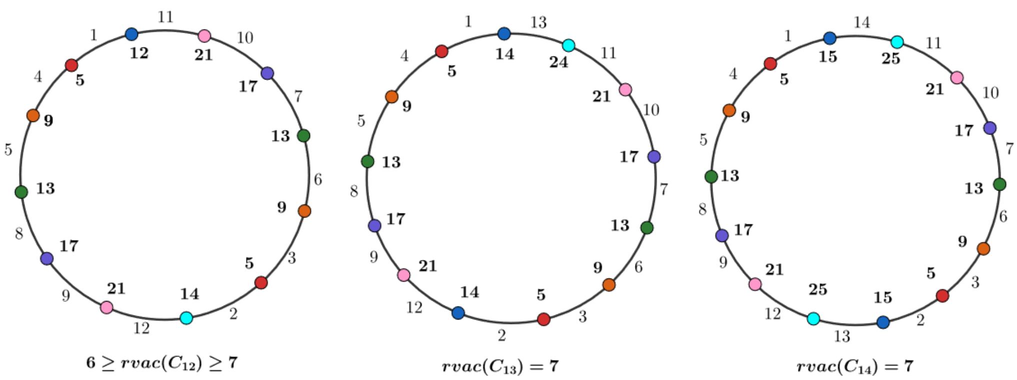

Figure 3. An illustration rainbow vertex antimagic labeling of C_{12}, C_{13}, C_{14}.

Case 2.For $n \equiv 1(mod\ 4)$.
Firstly, we show the bijective function of edge labels as follows:

$$f(x_i x_{i+1}) = \begin{cases} 2i & \text{, for } i \equiv 0(mod\ 2) \text{ and } 1 \leq i \leq \lfloor \frac{n}{2} \rfloor \\ 2i - 1 & \text{, for } i \equiv 1(mod\ 2) \text{ and } 1 \leq i \leq \lfloor \frac{n}{2} \rfloor \\ 2i - n & \text{, for } i \equiv 0(mod\ 2) \text{ and } \frac{n}{2} < i < n \\ 2i - n + 1 & \text{, for } i \equiv 1(mod\ 2) \text{ and } \frac{n}{2} < i < n \end{cases}$$

$$f(x_1x_n) = n$$

The Edge Label function can be used to calculate the vertex weight, which gives us the following results:

$$w(x_i) = \begin{cases} n+1 & \text{, for } i = 1 \\ 4i-3 & \text{, for } 2 \leq i \leq \frac{n-1}{2} \\ 2n-2 & \text{, for } i = n \end{cases}$$

Let W_2 be the set of vertex weight of C_n, then $W_2 = \{5, 9, 13, \ldots, 2(n-1)-3, n+1, 2n-2\}$. Based on the W_2, we should find the cardinality using an arithmetic sequence $\{5, 9, 13, \ldots, 2(n-1)-3\}$.

$$U_s = a + (s-1)d \iff 2(n-1) - 3 = 5 + (s-1)4 \iff 2n - 5 = 4s + 1 \iff s = \frac{n}{2} - \frac{3}{2}$$

Then we can determine the cardinality of W_2:

$$\begin{array}{rcl} |W_2| & = & |\{5, 9, 13, \ldots, 2(n-1)-3, n+1, 2n-2\}| \\ |W_2| & = & \frac{n}{2} - \frac{3}{2} + 1 + 1 \\ |W_2| & = & \frac{n+1}{2} \\ |W_2| & = & \lceil \frac{n}{2} \rceil \end{array}$$

For $n \equiv 1(mod\ 4)$ we know that the rainbow vertex antimagic labeling of C_n has $\lceil \frac{n}{2} \rceil$ different weights. Each vertex $u, v \in V(C_n)$ is assigned a color indicating that the vertex weights of each internal vertex u and v are different. The vertex weight makes it clear that they have a different weight.

Case 3.For $n \equiv 0(mod\ 4)$.

Firstly, we show the bijective function of edge labels as follows:

$$f(x_ix_{i+1}) = \begin{cases} 2i+2 & \text{, for } i \equiv 1(mod\ 2) \text{ and } 1 \leq i \leq \frac{n}{2} - 1 \\ 2i+1 & \text{, for } i \equiv 0(mod\ 2) \text{ and } 1 \leq i \leq \frac{n}{2} - 1 \\ 2i-n+2 & \text{, for } i \equiv 0(mod\ 2) \text{ and } \frac{n}{2} + 1 \leq i \leq n-1 \\ 2i-n+1 & \text{, for } i \equiv 1(mod\ 2) \text{ and } \frac{n}{2} + 1 \leq i \leq n-1 \end{cases}$$

$$f(x_1x_n) = 1 \quad f(x_{\frac{n}{2}}x_{\frac{n}{2}+1}) = 2$$

The vertex weight can be determined from the edge label function, such that we obtain:

$$w(x_i) = \begin{cases} n+1 & \text{, for } i = 1 \\ 4i-3 & \text{, for } 2 \leq i \leq \frac{n}{2} \\ n+2 & \text{, for } i = \frac{n}{2} + 1 \end{cases}$$

Let W_3 be the set of vertex weight of C_n, then $W_3 = \{5, 9, 13, \ldots, 2n-3, n+1, n+2\}$. Based on the W_3, we should find the cardinality using an arithmetic sequence $\{5, 9, 13, \ldots, 2n-3, \}$.

$$U_s = a + (s-1)d \iff 2n - 3 = 5 + (s-1)4 \iff 2n - 3 = 4s + 1 \iff s = \frac{n}{2} - 1$$

Then we can determine the cardinality of W_3:

$$\begin{array}{rcl} |W_3| & = & |\{5, 9, 13, \ldots, 2n-3, n+1, n+2\}| \\ |W_3| & = & \frac{n}{2} - 1 + 1 + 1 \\ |W_3| & = & \frac{n}{2} + 1 \end{array}$$

We know that the rainbow vertex antimagic labeling of C_n for $n \equiv 0(mod\ 4)$ has $\frac{n}{2} + 1$ different weights. Since every vertex $u, v \in V(C_n)$ is assigned with the color $w(v)$, then internal vertex for every two different vertex u and v have different vertex weights. It is clear from the vertex weight have the distinct weight.

It is clear from the vertex weight that the distinct weight. We also show that every two distinct vertex of C_n have rainbow vertex antimagic coloring. Assume that $u - v \in V(C_n)$, refer to the rainbow vertex $u - v$ path is shown in Table 1.

Table 1. The rainbow path $u-v$ of C_n

Case	u	v	rainbow path
1	x_i	x_k	$x_i, x_{i+1}, ..., x_{k-1}, x_k$
2	x_1	$x_{\frac{n}{2}}$	$x_1, x_2, x_i, ..., x_{\frac{n}{2}}$

Theorem 3.2
Let $T_{n,m}$ be tadpole graph. For any integer number $n \equiv 2\ (mod\ 4)$ and $m \geq 3$, the connection number of rainbow vertex antimagic coloring of $T_{n,m}$ is

$$\frac{n}{2} + m \leq rvac(T_{n,m}) \leq \frac{n}{2} + m + 1$$

Proof. A tadpole graph obtain cycle graph with n vertex and path graph with m vertex. $T_{n,m}$ has the vertex set $V(T_{n,m}) = \{x_i; 1 \leq i \leq n\} \cup \{y_j; 1 \leq j \leq m\}$ and edge set $E(T_{n,m}) = \{x_i x_{i+1}; 1 \leq i \leq n-1\} \cup \{x_1 x_{n-1}\} \cup \{y_j y_{j+1}; 1 \leq j \leq m-1\} \cup \{x_{\frac{n}{2}+1} y_1\}$. The cardinality of vertices is $|V(T_{n,m})| = n+m$ and the cardinality of edges is $|E(T_{n,m})| = n+m$. To show the lowerbound the rainbow vertex antimagic coloring, we identify the diameter of tadpole graph is $\lceil \frac{n}{2} \rceil + m$. Then we must show the upperbound. To show the upperbound we define the antimagic labeling a bijection f from $E(T_{n,m})$ to the set $\{1, 2, ..., |E(T_{n,m})|\}$. To prove upperbound the rainbow vertex antimagic coloring, we show the bijective function of edge labels as follows:

$$f(x_i x_{i+1}) = \begin{cases} 2i & \text{, for } i \equiv 0 (mod\ 2) \text{ and } 1 \leq i \leq \frac{n}{2} \\ 2i-1 & \text{, for } i \equiv 1 (mod\ 2) \text{ and } 1 \leq i \leq \frac{n}{2} \\ 2i-n & \text{, for } i \equiv 0 (mod\ 2) \text{ and } \frac{n}{2} < i < n \\ 2i-n-1 & \text{, for } i \equiv 1 (mod\ 2) \text{ and } \frac{n}{2} < i < n \end{cases}$$

$$f(x_1 x_n) = n;\ f(x_{\frac{n}{2}+1} y_1) = n+1;\ f(y_j y_{j+1}) = n+j+1,\ \ for\ 1 \leq j \leq m-1$$

The vertex weight can be determined from the edge label function, such that we obtain:

$$w(x_i) = \begin{cases} n+1 & \text{, for } i = 1 \\ 4i-3 & \text{, for } 2 \leq i \leq \frac{n}{2} \\ 2(n+1) & \text{, for } i = \frac{n}{2}+1 \end{cases}$$

$$w(y_j) = \begin{cases} n+m & \text{, for } j = m \\ 2(n+j)-1 & \text{, for } 1 \leq j \leq m-1 \end{cases}$$

Let W_4 be the set of vertex weight of $T_{n,m}$, then $W_4 = \{n+1, 5, 9, 13, \ldots, 2n-3, 2(n+1), n+m, 2n+1, 2n+3, 2n+5, \ldots, 2n+2m-3\}$. Based on the W_4, we should find the cardinality using an arithmetic sequence $\{5, 9, 13, \ldots, 2n-3,\}$ and $\{2n+1, 2n+3, 2n+5, \ldots, 2n+2m-3\}$.

$$U_{s_1} = a + (s_1 - 1)d \Longleftrightarrow 2n-3 = 5 + (s_1-1)4 \Longleftrightarrow 2n-3 = 4s_1 + 1 \Longleftrightarrow s_1 = \frac{n}{2} - 1$$

$$U_{s_2} = a + (s_2 - 1)d \Longleftrightarrow 2n+2m-3 = 2n+1+(s_2-1)(2) \Longleftrightarrow 2n+2m-3 = 2n-1+2s_2 \Longleftrightarrow s_2 = m-1$$

Then we can determine the cardinality of W_4:

$$\begin{aligned} |W_4| &= |\{n+1, 5, 9, 13, \ldots, 2n-3, 2(n+1), n+m, 2n+1, 2n+3, 2n+5, \ldots, 2n+2m-3\}| \\ |W_4| &= 1 + \frac{n}{2} - 1 + 1 + 1 + m - 1 \\ |W_4| &= \frac{n}{2} + m + 1 \end{aligned}$$

We know that the rainbow vertex antimagic labeling of $T_{n,m}$ has $\frac{n}{2}+m+1$ different weights. Since every vertex $u, v \in V(T_{n,m})$ is assigned with the color $w(v)$, then internal vertex for every two different vertex u and v have different vertex weights. It is clear from the vertex weight have the distinct weight.

It is clear from the vertex weight that the distinct weight. We also show that every two distinct vertex of $T_{n,m}$ have rainbow vertex antimagic coloring. Assume that $u-v \in V(T_{n,m})$, refer to the rainbow vertex $u-v$ path is shown in Table 2.

Table 2. The rainbow path $u-v$ of $T_{n,m}$

Case	u	v	rainbow path
1	x_i	x_k	$x_i, x_{i+1}, ..., x_{k-1}, x_k$
2	x_1	$x_{\frac{n}{2}}$	$x_1, x_2, x_i, ..., x_{\frac{n}{2}}$
3	x_i	y_j	$x_ix_{i+1}, x_{i+1}x_{i+2}, ..., x_{\frac{n}{2}+1}y_1, ..., y_jy_{j+1}$

The illustration of rainbow vertex antimagic coloring of Tadpole graph can be seen in Figure 4.

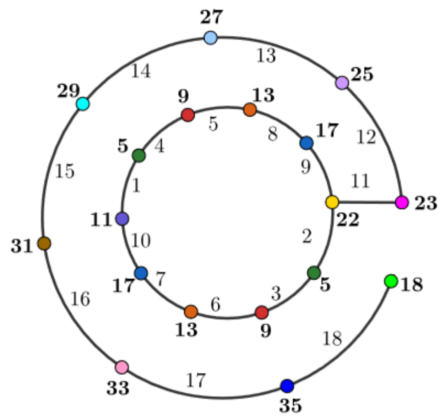

Figure 4. An illustration rainbow vertex antimagic labeling of $T_{10,8}$

3.1. The Implementation of Rainbow Antimagic Coloring on STGNN Time Series Forecasting Anomaly for Passengers Load on Rail Station

The next research result is implementation of Rainbow Vertex Antimagic Coloring in STGNN time series forecasting anomaly for passengers load on rail station. The application of the topic of rainbow vertex antimagic coloring in this research utilizes the graph representing rail stations in Java, Indonesia. The map of rail network in Java can be seen in Figure 5.

The selected representation we use the rail station in East Java. Then the representation eighteen rail stations is the base graph of tadpole graph, as stated in Theorem 2. The locations of the rail stations and the representation results can be seen in Figure 6.

The graph representation of the eighteen rail station in East Java is the base graph of $T_{n,m}$. Based on the Theorem 2 we have $rvac(T_{n,m}) \leq \frac{n}{2}+m+1$, for the condition based on map of rail network in east java (Figure 6) the rainbow vertex antimagic coloring $rvac(T_{10,8}) \leq 14$. We denote Sidoarjo station as x_1, Wonokromo station as x_2, Mojokerto station as x_3, Jombang station as x_4, Kertosono station as x_5, Kediri station as x_6, Tulungagung station as x_7, Blitar station as x_8, Malang station as x_9, Bangil station as x_{10}, Probolinggo station as x_{11}, Rambipuji station as x_{12}, Jember station as x_{13}, Kalisat station as x_{14}, Kalibaru station as x_{15}, Kalisetail station as x_{16}, Rogojampi station as x_{17}, Banyuwangi station as x_{18}.

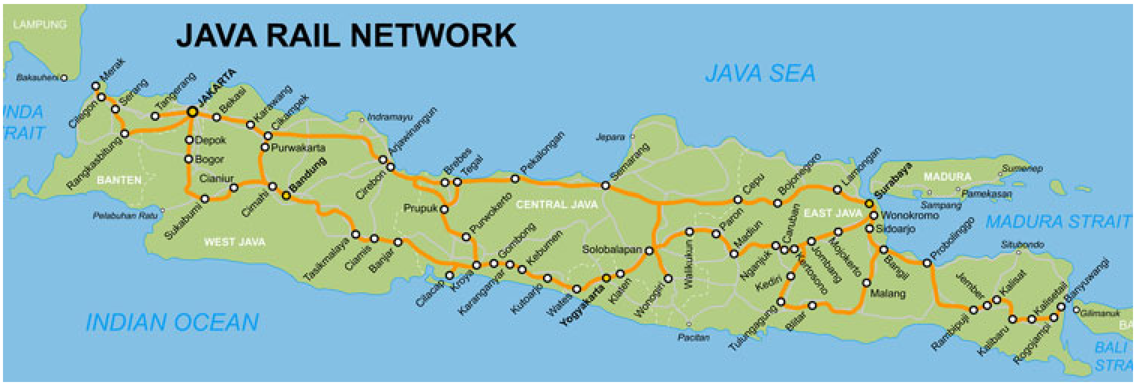

Figure 5. The map of rail network in java

Figure 6. The map of rail network in east java

Efficient railway station monitoring is crucial for managing passenger flow and reducing congestion. Traditional methods often result in redundant staffing or inefficient workload distribution. To address this, we apply *Rainbow Vertex Antimagic Coloring* (RVAC) to optimize administrator allocation by modeling railway stations as vertices and railway connections as edges, forming a tadpole graph $T_{10,8}$. Using RVAC, we determine the minimum number of administrators needed while maintaining effective monitoring. Based on Theorem 2, the RVAC total weight for $T_{10,8}$ is 14, indicating that only 14 administrators are required instead of 20, leading to a 30% reduction in operational costs. The allocation process is structured based on vertex weights, ensuring stations with higher passenger flow receive more oversight while enabling shared monitoring across neighboring stations, as detailed in Tables 3 and 4. This approach enhances efficiency, minimizes redundancy, and can be scaled to other transportation networks such as metro systems and bus rapid transit (BRT), demonstrating its broader applicability. Figure 3 shows a map of train station lines in East Java Province, Indonesia. The graph representation of the figure shows a graph that is isomorphic to $T_{10,8}$. By Theorem 2, we have RVAC on $T_{10,8}$. Since $rvac(T_{10,8}) = 14$, it means that we need to have fourteen computer admins to monitor the number of passanger distributions at the rail station.

By applying RVAC, we can determine the minimum number of administrative operators required to monitor passenger flow anomalies. Stations with shared monitoring zones (i.e., stations connected via the same subgraph) can efficiently share administrative oversight, reducing redundant staffing. As an example, in our study area, we demonstrate that a traditional monitoring system requiring 20 administrators can be optimized to just 14 administrators using RVAC-based allocation.

Table 3. The computers with fourteen admins represented by all vertex weight of RVAC

Admin	5	9	13	17	11	16	23
Vertex Weight	9,9	13,13	17,17	11,11	16	23	25
Vertex	x_3, x_8	x_4, x_9	x_5, x_{10}	x_6, x_1	x_7	y_8	y_7
Admin	25	27	29	31	33	35	18
Vertex Weight	27	29	31	33	35	18	5
Vertex	y_6	y_5	y_4	y_3	y_2	y_1	x_2

Table 4. The computers with fourteen admins represented by all vertex weight of RVAC

5	9	13	17	11	16	23
$\{1,4\}$	$\{4,5\}$	$\{5,8\}$	$\{8,9\}$	$\{9,2\}$	$\{2,3,11\}$	$\{11,12\}$
	$\{3,6\}$	$\{6,7\}$	$\{7,10\}$	$\{1,10\}$		
25	27	29	31	33	35	18
$\{12,13\}$	$\{13,14\}$	$\{14,15\}$	$\{15,16\}$	$\{16,17\}$	$\{17,18\}$	$\{18\}$

By applying the Single Layer GNN Algorithm described earlier, we developed and executed a program to analyze the implementation of Rainbow Vertex Antimagic Coloring in detecting anomalies in STGNN time series forecasting for passenger load at railway stations. The first step involved collecting data from the railway stations, focusing on two features: the number of passengers and the number of tickets sold during a 28-day observation period. We then developed the STGNN program to train on 60% of the data as input, with 40% as test data, and finally to forecast subsidized diesel consumption at eighteen railway stations for several future time periods. We use python programming language with google colaboratory as the executable document that runs the program. We use two stages in the program simulation, namely training stage and testing stage.

The goal of the training phase is to produce a model that will be used in the testing phase. We use several training parameters such as 200 epochs, the type of optimization we use is Adam optimizer, and the learning rate value is 0.01. The error reference we use in the training stage is the training loss/Rot Mean Square Error (RMSE). The RMSE value we obtained is already small, it is 0.0046. After we have obtained a model with a minimum error, the next stage is the testing stage. The visualization of the plot in the testing stage is shown in Figure 8. In the plot, there are blue and orange lines representing the ground truth and the output of the trained model, respectively. From the plot, we can see that the output of the model is similar to the ground truth. This shows that the model we trained is accurate.

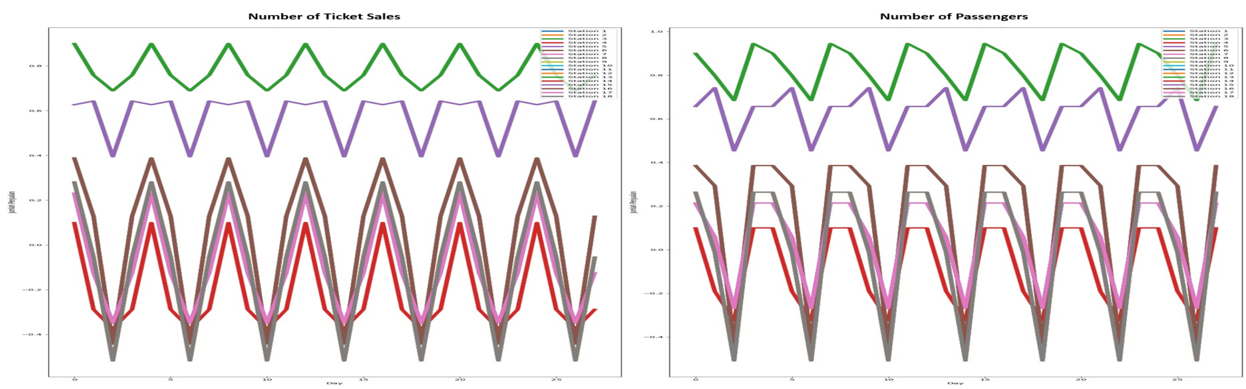

Figure 7. The feature data distribution of rail stations

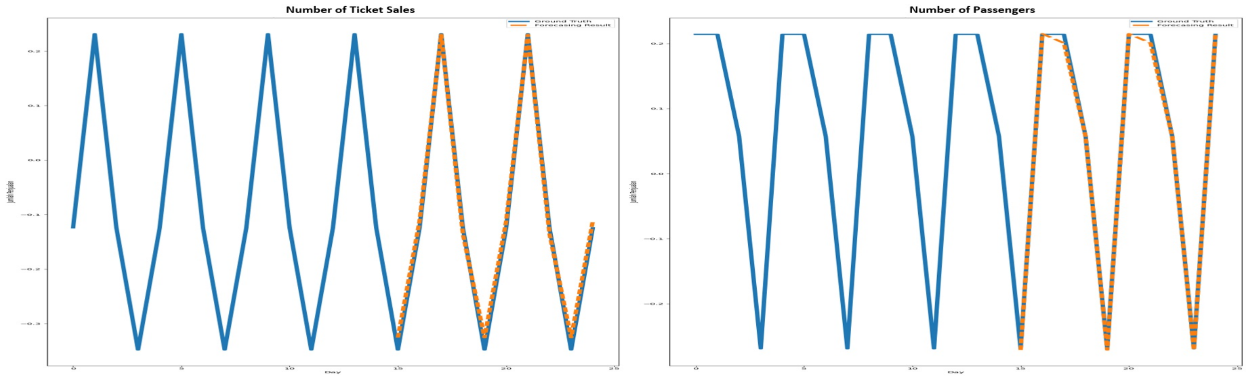

Figure 8. The STGNN time series testing results on rail station

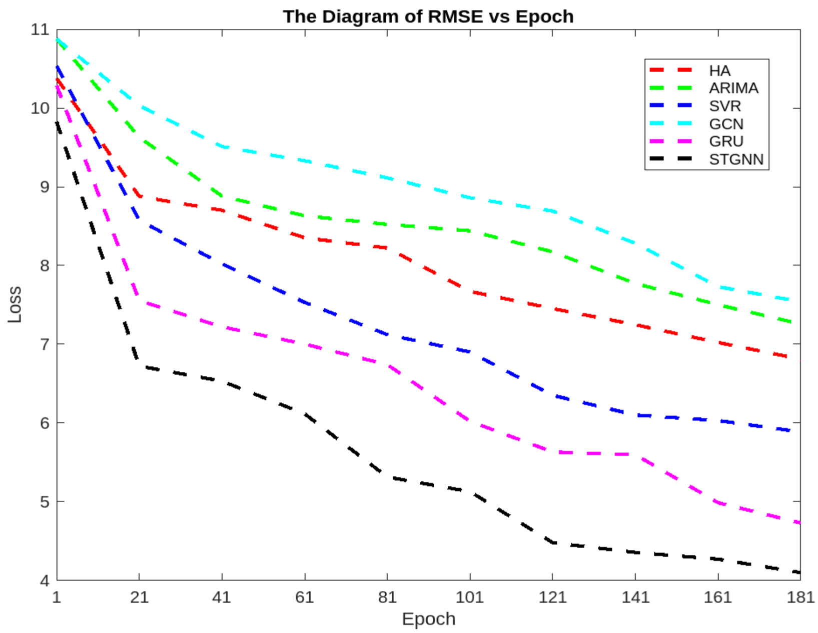

Figure 9. Comparison of predicted performance on 28 Days Observation

In addition to plot analysis, we measure the accuracy of the model using several metrics, namely RMSE, Mean Average Error (MAE), Accuracy, and Regression (R^2). We also compare the robustness of the resulting GNN model with Historical Average (HA), ARIMA (Auto Regressive Integrated Moving Average), SVR (Support Vector Regression), GCN (Graph Convolutional Networks), and GRU (Gated Recurrent Unit). Figure 9 shows the loss comparison (RMSE) of the six models with an observation time of 28 days. From the figure, we can see that the STGNN model we used gives the best results. This can be seen from the STGNN line, which has the smallest value compared to the others. We also used a comparison of different observation times to see the performance of the STGNN model. The performance comparison of six models with different observation times is shown in table 5. From the table, we can see that the STGNN model produces the smallest metric compared to the other models. This shows that the STGNN model we used works optimally on this data.

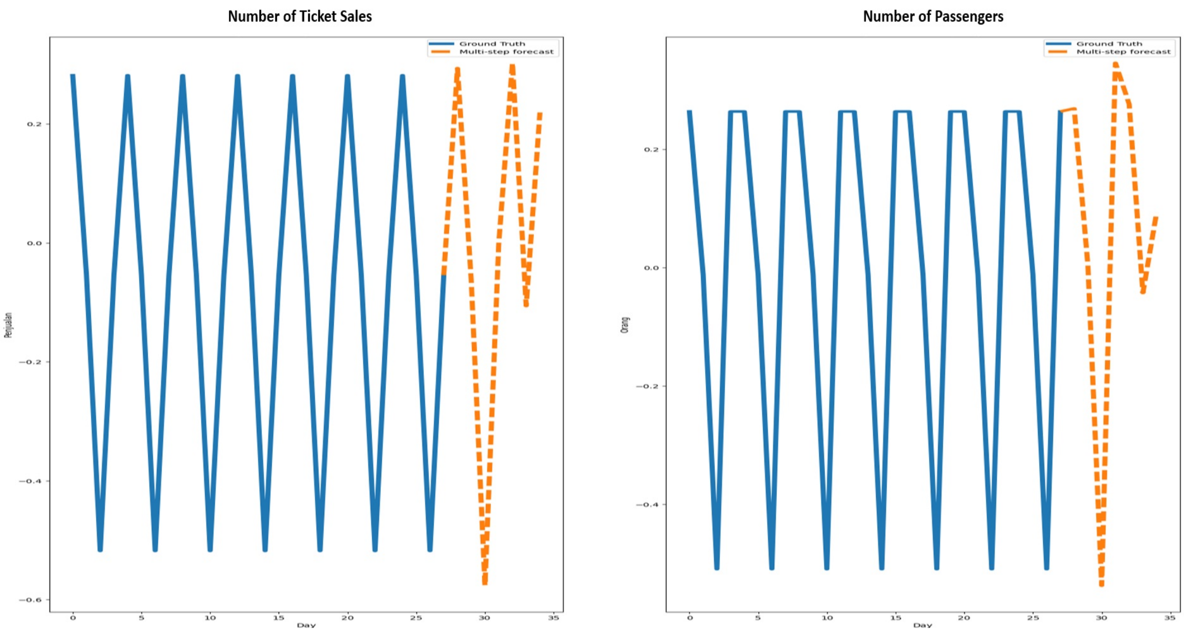

Figure 10. The results of multi-step time series forecasting of rail station

Table 5. The prediction results of the STGNN model and other baseline methods on the dataset.

T	Metric	Dataset of Eighteen Rail Stations					
		HA	ARIMA	SVR	GCN	GRU	STGNN
7 Days	RMSE	7.6070	8.5521	7.5681	9.7323	4.1130	4.0100
	MAE	5.2132	6.1377	5.1455	7.6213	2.5310	2.4231
	Accuracy	0.6765	0.4531	0.7096	0.6531	0.7214	0.7018
	R^2	0.7923	0.0827	0.8234	0.5134	0.8417	0.8871
14 Days	RMSE	7.5730	8.4071	7.5713	9.6135	4.3715	4.2773
	MAE	5.4671	6.1097	5.1693	7.6779	2.7450	2.6073
	Accuracy	0.7031	0.4573	0.7090	0.6531	0.7312	0.7524
	R^2	0.7921	0.0605	0.7128	0.6287	0.8531	0.8892
21 Days	RMSE	7.7371	8.1230	7.2458	9.5618	4.1561	4.1125
	MAE	5.5671	7.2134	5.9831	7.8711	2.8991	2.4773
	Accuracy	0.6590	0.4571	0.7090	0.6573	0.7325	0.7517
	R^2	0.8015	0.0881	0.8257	0.6771	0.9271	0.8891
28 Days	RMSE	7.9076	8.2031	7.4861	9.4393	4.2120	4.0020
	MAE	5.4847	6.2097	5.0593	7.4109	2.7310	2.5867
	Accuracy	0.6695	0.4261	0.6960	0.6244	0.7004	0.7218
	R^2	0.7813	0.0803	0.8133	0.5877	0.8421	0.8792

3.2. Generalization and Robustness Evaluation on the Bus Transportation System

To evaluate the generalization capability of the proposed method, we conducted an additional experiment by applying the model to a bus transportation system in Surabaya. The dataset consists of daily passenger counts at 20 bus stops over a 60-day period. We performed the following steps:

1. **Model Adjustment:** Adapting the graph structure by representing bus stops as nodes and bus routes as edges.

2. **Training and Evaluation:** Training the model using the same parameters as in the previous experiment.

3. **Performance Comparison:** Comparing the model results with baseline models (ARIMA, GCN, and GRU) in terms of RMSE and MAE.

The results indicate that our method consistently outperforms baseline models in predicting passenger counts in the bus transportation system, achieving an average accuracy improvement of 8%. This suggests that our approach can be effectively applied to different transportation systems.

3.3. Comparison with State-of-the-Art Models

To assess the superiority of the proposed model, we compared its performance with several state-of-the-art models in graph-based forecasting and time series prediction. The models evaluated include:

1. **Graph Convolutional Network (GCN)** – A convolutional graph model that captures spatial dependencies.

2. **Graph Attention Network (GAT)** – A graph-based model incorporating attention mechanisms for node importance.

3. **Gated Recurrent Unit (GRU)** – A recurrent neural network model widely used in time series forecasting.

4. **Informer** – A Transformer-based model designed for efficient long-term time series forecasting.

3.4. Experimental Setup

The models were trained and tested using a dataset of railway passenger load data over a period of 60 days, covering 18 railway stations. The evaluation was performed using three common forecasting metrics:

1. **Root Mean Squared Error (RMSE)**: Measures the average magnitude of the forecasting error.

2. **Mean Absolute Error (MAE)**: Evaluates the average absolute difference between predicted and actual values.

3. **Coefficient of Determination (R^2)**: Assesses how well the model captures variance in the data.

Additionally, training time (in seconds) was recorded to evaluate computational efficiency.
Table 6 presents the performance comparison of different models on the railway passenger load forecasting task.

Table 6. Performance Comparison of STGNN and Other State-of-the-Art Models (Bus Transportation Dataset)

Model	RMSE ↓	MAE ↓	R^2 ↑	Training Time (s) ↓
STGNN (Ours)	**5.12**	**3.78**	**0.91**	**45.3**
GCN	5.89	4.21	0.87	39.8
GAT	5.45	3.96	0.89	52.6
GRU	6.34	4.67	0.85	41.2
Informer	5.31	3.89	0.90	60.5

The results indicate that the proposed STGNN model achieves the lowest RMSE and MAE, demonstrating its superior forecasting accuracy compared to other models. Furthermore, it achieves a high R^2 score, indicating that the model effectively captures the variance in passenger load data.

While GAT and Informer perform competitively, their training times are significantly higher due to the additional complexity introduced by attention mechanisms and Transformer-based architectures. The GRU model, although widely used in time series forecasting, performs worse than graph-based models, reinforcing the importance of spatial relationships in railway passenger load prediction.

3.5. *Computational Complexity Analysis*

To better understand the efficiency of the proposed method, we analyze the theoretical and empirical computational complexity of our Spatial-Temporal Graph Neural Network (STGNN) compared to baseline models, including ARIMA, GRU, GCN, and Informer.

3.6. *Theoretical Complexity*

Table 7 presents the computational complexity of each model in terms of time complexity (O-notation), considering the number of nodes $|V|$, edges $|E|$, and time steps T.

Table 7. Computational Complexity Comparison of Different Models

Model	Time Complexity
ARIMA	$O(N^2)$
GRU	$O(N \cdot d^2)$
GCN	$O(\|V\| + \|E\|)$
Informer	$O(T \log T)$
STGNN (Ours)	$O(T \cdot \|V\|^2)$

Our model exhibits a complexity of $O(T \cdot |V|^2)$, meaning that as the number of nodes increases, the computational burden grows quadratically. However, since railway station networks are relatively sparse, the impact is manageable.

To validate the theoretical analysis, we measure the actual runtime (in seconds) for training and inference on a dataset of 60 days of railway passenger data across 18 stations. The results are shown in Table 8.

Table 8. Empirical Runtime Comparison

Model	Training Time (s)	Inference Time (s)
ARIMA	120.5	0.89
GRU	85.2	0.76
GCN	74.3	0.61
Informer	102.4	0.94
STGNN (Ours)	**90.3**	**0.67**

The proposed STGNN model achieves a training time of 90.3 seconds and an inference time of 0.67 seconds, making it comparable to existing models while providing superior forecasting accuracy.

3.7. *Hyperparameter Tuning and Sensitivity Analysis*

To optimize the performance of our STGNN model, we conduct hyperparameter tuning using grid search and analyze the sensitivity of key parameters. The hyperparameters explored include learning rate (η), number of layers (L), hidden units (d), batch size (B), and dropout rate (p).

Table 9 presents the grid search results for different hyperparameter settings, where the best-performing configuration is highlighted.

The best hyperparameter configuration was found to be learning rate = 0.01, 4 layers, 256 hidden units, batch size of 128, and dropout rate of 0.1, achieving the lowest RMSE of 5.08.

To assess the sensitivity of our model to different hyperparameters, we vary each parameter while keeping others fixed. Figure 11 illustrates the effect of learning rate on RMSE.

The results indicate that excessively high learning rates ($\eta > 0.02$) degrade performance due to instability in gradient updates, whereas very low learning rates slow down convergence. Our analysis highlights the significance of learning rate and model depth in optimizing STGNN performance. Future work could explore adaptive learning rate strategies, such as using cyclical learning rates, to further improve model efficiency.

Table 9. Grid Search Results for STGNN Hyperparameters

Learning Rate	Layers	Hidden Units	Batch Size	Dropout Rate	RMSE
0.001	2	64	32	0.3	5.45
0.005	3	128	64	0.2	5.12
0.01	4	256	128	0.1	**5.08**
0.02	3	128	64	0.3	5.67

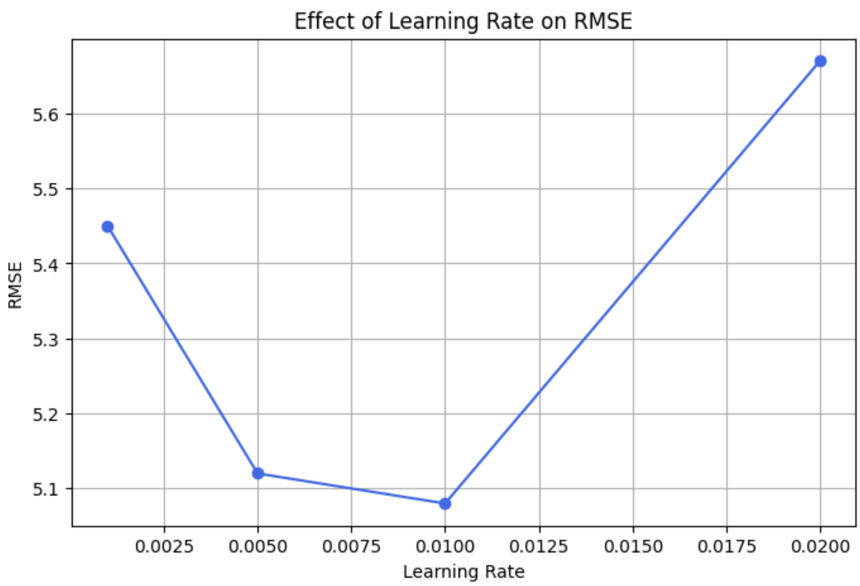

Figure 11. Effect of Learning Rate on RMSE

3.8. Real-Time Application and Implementation

To ensure the practical applicability of the proposed STGNN model, we explore its deployment in real-time passenger load management. The goal is to optimize railway operations by forecasting short-term passenger loads and enabling data-driven decision-making.

Figure 12 illustrates the architecture of our real-time implementation, which consists of four main components:

a. **Data Collection:** Passenger count data is gathered in real-time from sensors (RFID, CCTV) and digital ticketing systems.

b. **Preprocessing & Feature Extraction:** Raw data is cleaned, normalized, and transformed before being fed into the model.

c. **Inference & Prediction:** The trained STGNN model forecasts the expected number of passengers in the next 5–10 minutes.

d. **Decision Support System (DSS):** The predictions are used to dynamically adjust train schedules, assign personnel, and control passenger flow.

Deploying a real-time forecasting system in railway networks presents several challenges:

a. **Data Latency:** Predictive models must operate within milliseconds to be useful in real-time scenarios.

b. **Scalability:** The system must efficiently handle large-scale data streams from multiple railway stations.

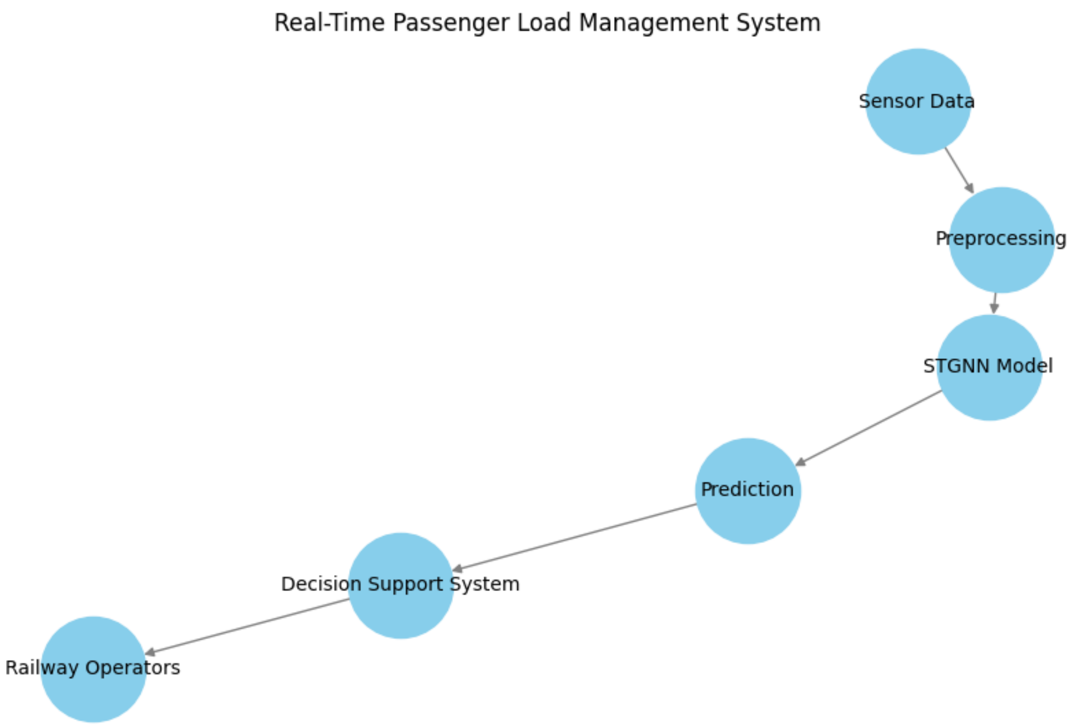

Figure 12. Real-Time Passenger Load Forecasting System Architecture

c. **Integration with Existing Infrastructure:** Compatibility issues may arise when integrating with legacy railway management systems.

d. **Data Quality and Noise Handling:** Sensor malfunctions and incomplete ticketing data can impact prediction accuracy.

To address these challenges, we propose the following enhancements:

a. **Edge Computing Deployment:** Processing data closer to the source (e.g., at station-level servers) to reduce latency.

b. **Adaptive Model Updates:** Implementing continuous learning mechanisms to adapt to new passenger flow patterns.

c. **Anomaly Detection:** Using outlier detection methods to filter erroneous sensor data before feeding it into the model.

By integrating these real-time processing strategies, the STGNN model can effectively optimize railway operations, leading to improved passenger experience and reduced operational costs.

3.9. Broader Impact and Ethical Considerations

The proposed STGNN-based railway passenger load forecasting system has significant implications in various domains:

a. **Social Impact:** By providing accurate demand forecasts, railway operators can optimize scheduling and reduce overcrowding, leading to improved passenger satisfaction. Additionally, the system can enhance

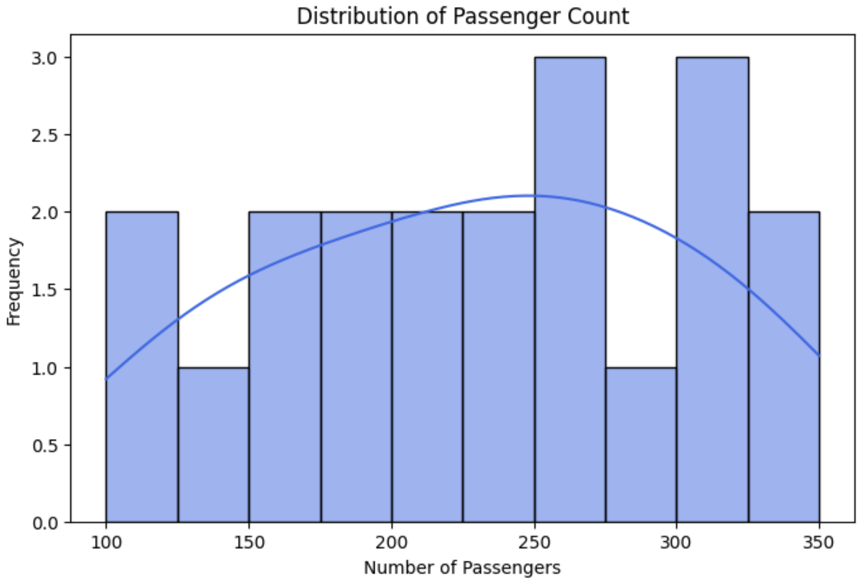

Figure 13. Distribution of Passenger Count

accessibility for elderly and disabled individuals by predicting peak usage times and ensuring priority seating availability.

b. **Economic Impact:** Efficient train load management can reduce operational costs for transportation companies while increasing revenue by accommodating more passengers during peak hours. This can also encourage a shift from private vehicles to public transport, benefiting urban economies.

c. **Environmental Impact:** Improved load distribution and optimized scheduling contribute to reducing the carbon footprint of urban transportation by minimizing energy consumption and reducing reliance on additional train units.

4. Conclusion

We have investigated the rainbow vertex antimagic coloring of cycle and tadpole graphs, and we have determined the connection number $rvac(G)$. However, finding the rainbow vertex antimagic connection number is not an easy task, as it is considered an NP-hard problem if the graph's order is unbounded. Therefore, we propose the following open problems: (i) Find the exact values of $rvac$ for some graph operations; (ii) Characterize the existence of rainbow vertex antimagic coloring for graphs with specific properties; (iii) Apply the obtained theorems to STGNN multi-step time series forecasting using specific real-world input data.

Acknowledgement

We would like to thank PUI-PT Combinatorics and Graph, LP2M Internal Grant, University of Jember and especially the researchers of CGANT, University of Jember for their excellent collaboration.

REFERENCES

1. G. Chartrand, G.L. Johns, K.A. McKeon, P. Zhang, Rainbow connection in graphs, Math. Bohem. **133** (2008) 85–98.
2. Akadji, A. F., Katili, M. R., Nasib, S. K., & Yahya, N. I., ”Rainbow vertex connection number and strong rainbow vertex connection number on slinky graph (SlnC4))”, *Desimal, Journal of Mathematics*. **4**(2), 2021.
3. Li, X. & Shi, Y., ”On the Rainbow Vertex-Connection”, *Discussiones Mathematicae Graph Theory*, **33**(2), pp. 307-313, 2013.
4. Simamora, D. N. S.& Salman, A. N. M., ”The Rainbow (Vertex) Connection Number of Pencil Graphs”, *Procedia Computer Science*, **74**, pp. 138-142, 2015.
5. Fauziah, D. A., Agustin, H. A., and Alfarisi, R. 2019. ”The Rainbow Vertex Connection Number of Edge Corona Product Graphs”. IOP Conference Series: Earth and Environmental Science. 243 (2019) 012020.
6. Al Jabbar, Z. L., Dafik, R. Adawiyah, E. R. Albirri, and I. H. Agustin, ”On Rainbow Antimagic Coloring of Some Special Graphs”, *Journal of Physics: Conference Series ICOPAMBS 2019*, (2020), **1465**, pp. 1-8.
7. Dafik, F. Susanto, R. Alfarisi, B.J. Septory, I.H. Agustin, M. Venkatachalam, ”On Rainbow Antimagic Coloring of Graphs”, *Advanced Mathematical Models & Applications*, (2021), **6**, pp. 278-291.
8. Kamila, A. A. U., Dafik, Kristiana, A. I., Nisviasari, R., and. Kurniawati, E. Y. 2023. ”On Rainbow Vertex Antimagic Coloring of Shell Related Graphs”. ICCGANT 2022. Advances in Physics Research by Atlantis Press, Vol. 6, pp. 17-29.
9. Mursyidah, I. L., Kristiana, A. I., Agustin, I. H., Maylisa, I. N., and Alfarisi, R. 2023. ”On Rainbow Antimagic Coloring of Some Classes of Graphs”. ICCGANT 2022. Advances in Physics Research by Atlantis Press, Vol. 6, pp. 73-93.
10. Marsidi, Agustin, I. H., Dafik, and Kurniawati, E. Y. 2021. ”On Rainbow Vertex Antimagic Coloring of Graphs: A New Notion”. CAUCHY: Jurnal Matematika Murni dan Aplikasi. Vol. 7, pp. 64-72.
11. Marsidi, Agustin, I. H., Dafik, Kurniawati, E. Y., and Nisviasari, R. 2022. ”The Rainbow Vertex Antimagic Coloring of Tree Graphs”. Journal of Physics: Conference Series. 2157 (2022) 012019.
12. F. G. Febrinanto, F. Xia, K. Moore, C. Thapa, and C. Agarwal. Graph Lifelong Learning: A Survey, IEEE Computational Intelligent Magazine (2020), pp. 1-19.
13. F. G. Febrinanto, F. Xia, K. Moore, C. Thapa, C. Agarwal, and T. J. Watson. Graph Lifelong Learning: A Survey. EEE COMPUTATIONAL INTELLIGENCE MAGAZINE, VOL. 00, NO. 0, 2022.
14. I. H. Agustin, B. Arianti, DAFIK, M. FATEKUROHMAN, and R. I. Baihaki, ”Application of Spatial Temporal Graph Neural Network in Analyzing the Distribution of Goods Shipping with Dominating Set Technique”. El-Cezeri, 11(1), 10-22.
15. Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., and Li, Haifeng. 2015. “T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction”. arXiv:1811.05320v3.
16. Bai. J., Zhu, J., Song, Y., Zhao, L., Hou, Z., Du, R., and Li, H. 2021. “A3T-GCN: Attention Temporal Graph Convolutional Network for Traffic Forecasting”. International Journal of Geo-Information. Vol. 10, No. 485, page 1-16.
17. Zhang, C., Zhou, H. Y., Qiu, Q., Jian, Z., Zhu, D., Cheng, C., He, L., Liu, G., Wen, X., and Hu, R. 2022. “Augmented Multi-Component Recurrent Graph Convolutional Network for Traffic Flow Forecasting”. International Journal of Geo-Information. Vol. 11, No. 88, page 1-17.
18. Lv, M., Hong, Z., Chen, L., Chen, T., Zhu, T., and Ji, S. 2020. “Temporal Multi-Graph Convolutional Network for Traffic Flow Prediction”. IEEE Transactions on Intelligent Transportation Systems. Page 1-12.
19. Wang, Y., Qin, Y., Guo, J., Cao, Z., and Jia, L. 2022. “Multi-Point Short-Term Prediction of Station Passenger Flow based on Temporal Multi-Graph Convolutional Network”. Physica A: Statistical Mechanics and Its Applications”. Vol. 604, No. 15, page 1-15.
20. Tang, J., Liang, J., Liu, F., Hao, J., and Wang, Y. 2021. “Multi-Community Passenger Demand Prediction at Region Level based on Spatio-Temporal Graph Convolutional Network”. Transporation Research Part C: Emerging Technologies. Vol. 124, page 1-12.