El Cezeri
Journal of Science and Engineering
ISSN: 2148-3736

Research Article

The Implementation of Spatial-Temporal Graph Neural Network in Analyzing the Distribution of Goods Shipping with Dominating Set Technique

Ika Hesti Agustin[1a,2], **Binti Arianti**[1b], **Mohamad Fatekurohman**[1c] **Dafik Dafik**[2d,3] and **Rifki Ilham Baihaki**[2e]

[1] Department of Mathematics, University of Jember, Jember, Indonesia
[2] PUI-PT Combinatorics and Graph, University of Jember, Jember, Indonesia
[3] Department of Postgraduate Mathematics Education, University of Jember, Jember, Indonesia

aikahesti.fmipa@unej.ac.id

DOI : 10.31202/ecjse.1289020
Received: 28.04.2023 Accepted: 17.01.2024
How to cite this article:
Ika Hesti Agustin, Binti Arianti, Mohamad Fatekurohman, Dafik Dafik, Rifki Ilham Baihaki, " The Implementation of Spatial-Temporal Graph Neural Network in Analyzing the Distribution of Goods Shipping with Dominating Set Technique", El-Cezeri Journal of Science and Engineering, Vol: 11, Iss:1, (2024), pp.(10-22).
ORCID: [a]0000-0001-7508-0760; [b]0009-0005-8125-1489. [c]0000-0003-3312-5558; [d]0000-0003-0575-3039; [e]0000-0002-2444-2829.

Abstract : A logistics service provider may face capacity problems due to distribution delays, resulting in goods accumulating at branches with unknown locations. To solve this problem, we will implement the Spatial-Temporal Graph Neural Network (STGNN) combined with the dominating set technique to predict the locations of these branches. The STGNN uses graph theory to represent the relationships between outlets in Indonesia. Simulation data on goods shipments across Indonesia are observed for 30 days, categorised as spatio-temporal data and fed into the STGNN. This process involves three stages: node embedding, training, and testing/forecasting. We implement several Artificial Neural Network (ANN) models with different hidden layer architectures. The results show that the best model of ANN is the cascade forward network and the MSE $1.6714\ 10^{-9}$.

Keywords : Spatial-Temporal Graph Neural Network, Goods Shipping, Logistics Services, Dominating Set.

1 Introduction

Expedition services have become important in many areas of life. This is due to the rapid development of technology and online businesses. Demand for delivery of goods by courier services from online businesses in Indonesia is the highest, accounting for 85.2% of all demand for delivery of goods [1]. Courier companies have branches all over Indonesia; each branch has a different distribution capacity for delivering goods, depending on the location, office area and number of employees [2]. The excess distribution capacity of goods delivery at the branch office often occurs, which will cause a problem. These conditions can cause delays in the distribution of goods delivery, so that there will be a build-up of goods in branches where the exact location is unknown [3]. Observations on simulation data of the types of goods that can be sent by courier services throughout Indonesia will be carried out regularly so that the data can be categorised as spatio-temporal [4]. The data are collected in time-series form, so this study is considered a Spatial-Temporal Graph Neural Network (STGNN). The STGNN method, equipped with a dominating set technique, is used to solve the problem of predicting the location of branches of freight forwarding companies that have excess distribution capacity for the delivery of goods [5].

Graph Neural Network (GNN) is a concept development of ANN, which is more effective if the data source is a non-Euclidean data domain represented in a network. We use it to solve problems in the distribution of goods delivery [6] [7] [8]. An Artificial Neural Network is a mathematical and computational model inspired by the structure and function of the human brain [8]. GNN is included in machine learning like ANN [9] [10]. Graph $G = (V, E)$, where V represents a vertex or an element, and E represents an edge or a connection. In GNN, the vertex of G may have some features in the same types and different. The network of branch offices of freight forwarding companies throughout Indonesia can be represented in graphs, with connections between branch offices viewed as edges of the graph [11] [12]. In contrast, branch office locations are viewed as vertices of graphs. GNN can extract information from a network as a graph, with each vertex used as a unit that can be updated and each weighted edge as a message delivery path [13].

The information that will be used as input in the GNN method is vertex features and weighted edges, which will be processed through three stages of node embeddings:

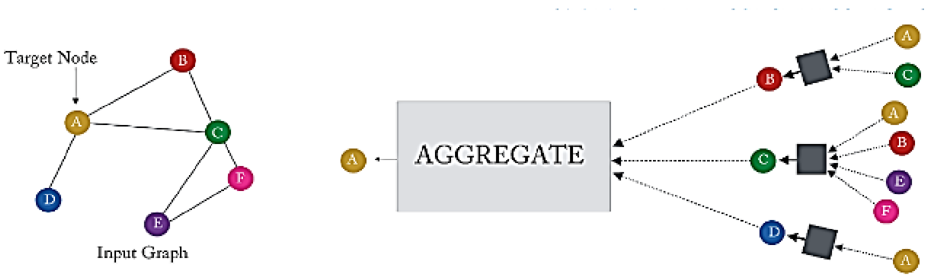


Figure 1: Message Passing and Aggregation Stage.

- message passing,
- message aggregates, and
- updates, so that the feature dimensions are getting lower.

The message passing stage, as illustrated in Figure 1 with point A as the target, will calculate the passing value for each vertex adjacent to vertex A, namely vertex B, C, and D. The passing values from vertex B, C, and D are obtained from the vertex adjacent to the three vertices, respectively are the vertex A, C, vertex A, B, E, F, and vertex A. The aggregate stage adds the passing values from vertex B, C, and D to obtain an embedding matrix, which will be the input to the ANN method. The embedding matrix reduces the dimensions of large vertex features so that each vertex has a smaller dimension. The indicator of the effectiveness of the embedding process is if similarity in the original graph represents similarity in the embedding space [13]. The study about GNN can be seen in [5] [14] [15].

A set of vertex S in G is the dominating set in G if some vertex in S dominates every vertex in G, or it can be said that a set of vertex S in G is the dominating set of G if every vertex in $V(G) - S$ is adjacent to some vertex in S [16] [17]. The dominating set of the spatial graph will be used to predict the location of forwarding service branch offices that experience excess distribution capacity for goods delivery. The study about dominating a set of graphs can be seen in [18] [19] [20] [21].

2 Experimental Methods

Here is the single-layer algorithm for learning the problem of excess capacity in the distribution of goods in a branch of a logistics company using STGNN combined with the dominating set technique.

Graph Neural Network Algorithm

Step 1 Prepare a graph $G(V, E)$ with order p

Step 2 Initialize the error tolerance value (ϵ) and learning rate value (α)

Step 3 Determine the adjacency matrix $(A_{(pxp.)}$)

Convert the relationships between a vertex in graph G into an adjacency matrix $(A_{p\times p})$ according to the rule in Equation 2

$$A_{ij} = \begin{cases} 1 & if\ c_i, c_j \in V(G)\ so\ that\ (c_i c_j) \in E(G), \\ 0 & if\ c_i, c_j \in V(G)\ but\ (c_i c_j) \notin E(G) \end{cases} \tag{1}$$

Step 4 Determining the loop adjacency matrix $(B_{p\times p})$

Adding the adjacency matrix $(A_{p\times p})$ from the data network with the identity matrix $(I_{p\times p})$

$$B = A + I \tag{2}$$

Description: B = loop adjacency matrix, A = adjacency matrix, I = identity matrix

Step 5 To convert the features of each point into a feature matrix, $H_{v_i}^{l-1}$ with a size of $(p \times q)$, where p is the number of vertex, q is the number of points and q is the number of features for each vertex.

$$H_{v_i}^{l-1} = [H_{v_1}^{l-1}\ H_{v_2}^{l-1}\ H_{v_3}^{l-1} \vdots H_{v_p}^{l-1}] \tag{3}$$

Description: $H_{v_i}^{l-1}$ = row vectors for features of each vertex

$H_{v_i}^{l-1} = [f_{i1}^{l-1}\ f_{i2}^{l-1}\ f_{i3}^{l-1}\ \cdots\ f_{iq}^{l-1}]$

$i = 1,\ 2,\ 3,\ \ldots,\ p$

$j = 1,\ 2,\ 3,\ \ldots,\ q$

l = iteration $0,\ 1,\ 2,\ 3,\ \ldots$

Thus, a matrix is obtained based on the feature data per point.

$$H_{v_i}^{l-1} = \left[f_{11}^{l-1}\, f_{12}^{l-1}\, f_{13}^{l-1} \;\cdots\; f_{1q}^{l-1} \;\; f_{21}^{l-1}\, f_{22}^{l-1}\, f_{23}^{l-1} \;\cdots\; f_{2q}^{l-1} \;\; f_{31}^{l-1} \vdots f_{p1}^{l-1} \;\; f_{32}^{l-1} \vdots f_{p2}^{l-1} \;\; f_{33}^{l-1} \;\cdots\; f_{3q}^{l-1} \vdots \ddots \vdots f_{p3}^{l-1} \;\cdots\; f_{pq}^{l-1} \right] \tag{4}$$

Description : $H_{v_i}^{l-1}$ = feature matrix
f_{pq}^{l-1} = matrix feature entries

Step 6 Normalizing the feature matrix

Normalizing the matrix $H_{v_i}^{l-1}$ using Equation 5 so that a new matrix $H_{v_i}^{l-1}$ is obtained, which will be the input in the GNN method.

$$x' = a + \frac{(x - x_{min}) \times (b - a)}{x_{max} - x_{min}} \tag{5}$$

Step 7 Initialize the initial weights

Initialize a weight matrix of size $(1 \times q)$ whose entries are random numbers in the range $0 - 1$

$$W^1 = [w_1\; w_2\; w_3\; \ldots\; w_q] \tag{6}$$

Description : W^1 = initial weight matrix
w_q = the weight in column-q

Step 8 Perform message passing

Calculate the passing value of each point using the following equation

$$m_{v_i}^l = W^l \cdot H_{v_i}^{l-1} \tag{7}$$

Description: W^l = row weight matrix at iteration to l

The passing values of each point are then organized into a column vector of size $(p \times 1)$

$$m_{v_i}^l = \left[m_{v_1}^l\; m_{v_2}^l\; m_{v_3}^l \vdots m_{v_p}^l \right] \tag{8}$$

Description: $m_{v_i}^l$ = message passing matrix at iteration to l

Step 9 Perform aggregate

Perform operations SUM on $m_{v_i}^1$ based on the nonzero elements of the matrix B

$$h_{v_i}^l = \sum_{i=1}^{p} m_{v_i}^l \times Bh_{v_i}^l = [h_{v_1}^l\; h_{v_2}^l\; h_{v_3}^l \vdots h_{v_p}^l] \tag{9}$$

Description: $h_{v_i}^l$ = matrix of node embeddings for each vertex

Step 10 Calculating the loss value (e)

The final stage of an iteration is closed by calculating the loss value (e)

$$e^l = \frac{\|h_{v_i}^l - h_{v_j}^l\|_\infty}{|E(G)|} \tag{10}$$

Description: e^l = loss value at iteration to l
(v_i, v_j) = two adjacent vertex
$\|h_{v_i}^l - h_{v_j}^l\|_\infty$ = $norm_\infty$ vertex passing value (v_i, v_j)
$|E(G)|$ = number of edges in G

Step 11 Checking whether the value of $e^l \leq \varepsilon$

If $e^l \leq \varepsilon$, then iteration **STOP** until this step
If $e^l \nleq \varepsilon$ then go to **Step 12**

Step 12 *Update* the feature matrix

Entering the next iteration needs to update the entries of the feature matrix $H_{v_i}^l$ using the formula

$$H_{v_i}^{l-1} = \frac{H_{v_i}^{l-2}}{\sum_{i=1}^{p} (H_{v_i}^{l-2})} \times h_{v_i}^{l-1} \tag{11}$$

So that we get

$$H_{v_i}^{l-1} = [f_{11}^{l-1}\, f_{12}^{l-1}\ \ldots f_{1q}^{l-1}\ f_{21}^{l-1}\, f_{22}^{l-1}\ \ldots f_{2q}^{l-1}\ \vdots f_{p1}^{l-1}\ \vdots f_{p2}^{l-1}\ \ddots\ \ldots\ \vdots f_{pq}^{l-1}] \tag{12}$$

Description : $H_{v_i}^{l-1}$ = new feature matrix
$H_{v_i}^{l-2}$ = feature matrix of the previous iteration
$h_{v_i}^{l-1}$ = embedding matrix of the previous iteration
f_{pq}^{l-1} = new feature matrix entry

Step 13 Update the weight matrix

Calculating the average of each column of the matrix $H_{v_i}^{l-1}$ that has been updated, then compile it in a matrix z_k, which will be used to update the weight matrix

$$z_k = [z_1\, z_2\ \ldots\ z_q] \tag{13}$$

$$W^l = W_q^l + \alpha \times z_k \times e^l \tag{14}$$

Description: z_k = average matrix of each column $H_{v_i}^{l+1}$
z_q = the average in column -q
We = new weight matrix
α = learning rate value

$$k = 1,\ 2,\ \ldots,\ q$$

Step 14 Perform Steps 8-11 until the value $e^l \le \varepsilon$

Step 15 Save the node embeddings matrix $(h_{v_i}^l)$

The node embedding matrix $(h_{v_i}^l)$ that has been obtained is stored to become the input value in the ANN method. If the data used is time series data, then the data for subsequent periods is processed through **Steps 5-15**.

Step 16ANN Method

The node embedding matrix that has been stored will be input data for the ANN model. The following process is training, testing, and forecasting using several models and architectures. We choose specific activation functions, and the best ANN model and architecture are obtained according to the smallest MSE value.

Examples of simulation data calculations using the GNN method from the first day to the 30 days to be used as the input of the ANN model.

1) Initialize the tolerance value $(\varepsilon = 10^{-5})$ and learning rate $(\alpha = 0,1)$;
2) Determining the feature matrix (H_{v_i}). Converts the sum of each type of goods in each city on the day into a feature matrix H_{v_i} with size (58×15).

   $[H_{v_i}^0 = [v_1\ v_2\ v_3 \vdots v_p]]$

   $[H_{v_i}^0 = [26\ 150\ 159\ \cdots\ 59\ \ 89\ 182\ 184\ \cdots\ 78\ \ 46\ \cdots\ 12\ \ 74\ \cdots\ 39\ \ 114\ \cdots\ 85\ \ \cdots\ \ddots\ \cdots\ \ 27\ \cdots\ 54];]$
3) Normalize the feature matrix (H_{v_i}). Normalizing the matrix using the rules in Equation 5, thus obtaining a new matrix H_{v_i}.

   $[H_{v_i}^0 = [0,24\ \ 0,70\ \ 0,73\ \ \cdots\ \ 0,51\quad 0,86\ \ 0,84\ \ 0,85\ \ \cdots\ \ 0,70\quad 0,44 \vdots 0,10\quad 0,36 \vdots 0,20\quad 0,51 \vdots 0,37\ \cdots\ \ddots\ \cdots\ \ 0,19 \vdots 0,46],\ i = 1,\ 2,\ 3,\ \ldots,\ 58]$
4) Convert the relationship between cities into an adjacency matrix (A) with a size (58×58).

   $[A = \left[0\,1\,1\ \cdots\ 0\ \ 1\,0\,1\ \cdots\ 0\ \ 1 \vdots 0\ \ 1 \vdots 0\ \ 0 \vdots 0\ \ \cdots\ \ddots\ \cdots\ \ 0 \vdots 0\right]]$
5) Adding the adjacency matrix (A) with the identity matrix (I) to obtain the matrix B. $[B = A + I = [1\,1\,1\ \cdots\ 0\ \ 1\,1\,1\ \cdots\ 0\ \ 1 \vdots 0\ \ 1 \vdots 0\ \ 1 \vdots 0\ \ \cdots\ \ddots\ \cdots\ \ 0 \vdots 1]]$

   The matrix B represents the relationship of each city to other cities and the relationship of each city to itself. The loop of each city to itself aims to keep the information in itself from being lost.
6) The initial weight value is a random number in the range 0 up to 1, which is converted into a matrix W^1 with size (1×15). $[W^1 = [0,05\ 0,05\ 0,05\ \cdots\ 0,05]]$
7) The iteration begins with the message-passing stage; in the first iteration, message-passing is done using Equation 7. $[m_{v_i}^l = W^l \cdot H_{v_i}^{l-1},\ i = 1,\ 2,\ 3, \ldots,\ 58]$

   Thus obtained, $[m_{v_i}^1 = [m_{v_1}^1\ m_{v_2}^1\ m_{v_3}^1 \vdots m_{v_{58}}^1] = [0,3217\ 0,5290\ 0,2469 \vdots 0,2269]]$
8) Determine *aggregate* (operation SUM) on $m_{v_i}^1$ based on the nonzero elements of matrix B.

$[h^1_{v_1} = 1,0976;\ h^1_{v_2} = 1,3952;\ h^1_{v_3} = 1,8585;\ldots;\ h^1_{v_{58}} = 0,5079]$
So that the node embeddings result in the matrix of the first iteration aggregate stage is obtained,

$$h^1_{v_i} = \left[h^1_{v_1}\ h^1_{v_2}\ h^1_{v_3} \vdots h^1_{v_{58}}\right] = \left[1,0976\ 1,3952\ 1,8585 \vdots 0,5079\right],\ i = 1,\ 2,\ 3,\ \ldots,\ 58$$

9) Each iteration ends by calculating the loss value (e) using Equation 10.

$$e^l = \frac{\|h^l_{v_i} - h^l_{v_j}\|_\infty}{|E(G)|},\ i,j = 1,\ 2,\ 3,\ 4$$

$$e^1 = \frac{\|h^1_{v_1} - h^1_{v_2}\| + \|h^1_{v_1} - h^1_{v_3}\| + \cdots + \|h^1_{v_{56}} - h^1_{v_{57}}\|}{135} = 1,0268$$

10) Checking value $e^1 \leq \varepsilon$?

$$e^1 \leq \varepsilon$$

$$1,0268 \leq 10^{-5}\ \text{(Wrong)}$$

Since the result is wrong, we proceed to the second iteration.

11) Update feature matrix (H_{v_i})
The second iteration begins with updating the matrix using Equation 11
$[H^{l-1}_{v_i} = \frac{H^{l-2}_{v_i}}{\sum_{i=1}^{p}(H^{l-2}_{v_i})} \times h^{l-1}_{v_i}]$
Thus obtained,

$$H^1_{v_i} = [H^1_{v_1}\ H^1_{v_2}\ H^1_{v_3} \vdots H^1_{v_{58}}] = [0,0406\ 0,1190\ 0,1237\ \cdots\ 0,0870\ \ 0,1135\ 0,1110\ 0,1116 \cdots$$

$$0,0923\ \ 0,1640 \vdots 0,0112\ \ 0,1341 \vdots 0,0223\ \ 0,1910 \vdots 0,0410\ \ \cdots\ \ddots\ \cdots\ \ 0,0715 \vdots 0,0515]$$

12) Calculating the average of each column of the matrix H that has been updated using the formula,
$[z_1 = 0,1400;\ z_2 = 0,1254;\ z_3 = 0,1147;\ldots;\ z_{15} = 0,1401]$
The values that have been obtained are then arranged in a matrix,
$[z_k = [z_1\ z_2\ z_3\ \cdots\ z_{15}] = [0,1400\ 0,1254\ 0,1147\ \cdots\ 0,1401]]$
Update the values in the weight matrix using Equation 14,
$[W^l = W^{l-1}_q + \alpha \times z_k \times e^{l-1}]\ [W^2 = W^1_q + \alpha \times z_k \times e^1]$
$\alpha = 0,1$ (adopting the learning rate in ANN)
Thus obtained,
$[W^2 = [W^2_1\ \ W^2_2\ \ W^2_3\ \ \ldots\ \ W^2_{15}]]\ [W^2 = [0,0644\ 0,0629\ 0,0618\ \ldots\ 0,0644]]$

13) Perform message passing for the second iteration using Equation 7,
$[m^l_{v_i} = W^l \cdot H^{l-1}_{v_i},\ i = 1,\ 2,\ 3,\ 4]$
Using the updated feature matrix and weight matrix, we obtain,

$$m^2_{v_i} = [m^2_{v_1}\ m^2_{v_2}\ m^2_{v_3} \vdots m^2_{v_{58}}] = [0,0694\ 0,0886\ 0,1177 \vdots 0,0323]$$

14) Aggregate for the second iteration using operations *SUM* on $m^2_{v_i}$ based on the nonzero elements of the matrix B.
$[h^2_{v_1} = 0,2757;\ h^2_{v_2} = 0,4880;\ h^2_{v_3} = 0,5922;\ldots;\ h^2_{v_{58}} = 0,1561]$
So that the node embeddings result matrix of the second iteration aggregate stage is obtained,

$$[h^2_{v_i} = [h^2_{v_1}\ h^2_{v_2}\ h^2_{v_3} \vdots h^2_{v_{58}}] = [0,2757\ 0,4880\ 0,5922 \vdots 0,1561],\ i = 1,\ 2,\ 3,\ \ldots,\ 58]$$

15) Calculating the loss value (e^l) for the second iteration using Equation 10.
$[e^l = \frac{\|h^l_{v_i} - h^l_{v_j}\|_\infty}{|E(G)|},\ i,j = 1,\ 2,\ 3,\ 4]\ [e^2 = \frac{\|h^2_{v_1} - h^2_{v_2}\| + \|h^2_{v_1} - h^2_{v_3}\| + \cdots + \|h^2_{v_{56}} - h^2_{v_{57}}\|}{135} = 0,5360]$

16) Iteration continues until $e^l \leq \varepsilon$ is true, with $\varepsilon = 10^{-5}$ as the predetermined error tolerance value.

17) Save the *node embeddings* matrix $(h^l_{v_i})$ obtained from the last iteration on day 30. is stored to be the input value in the ANN method.

Simulation of the ANN method

The architecture model that will be used for simulation is a feed-forward network, with input obtained from the results of node embeddings in point 14, one hidden layer with one neuron, and one output, as shown in Figure 2. and one output. Initialize the learning rate value $(\alpha = 0,1)$, target value $(t = 1)$, bias weight $(b = [0,1\ \ \ 0,1\ \ \ 0,1])$, and the edge weights between layers are
$v_1 = [0,1\ \ \ 0,2\ \ \ 0,3\ \ \ 0,1\ \ \ 0,2\ \ \ldots\ \ \ 0,1]$ (weight of the edge leading to z_1)
$v_2 = [0,2\ \ \ 0,3\ \ \ 0,1\ \ \ 0,2\ \ \ 0,3\ \ \ldots\ \ \ 0,2]$ (weight of the edge leading to z_2)

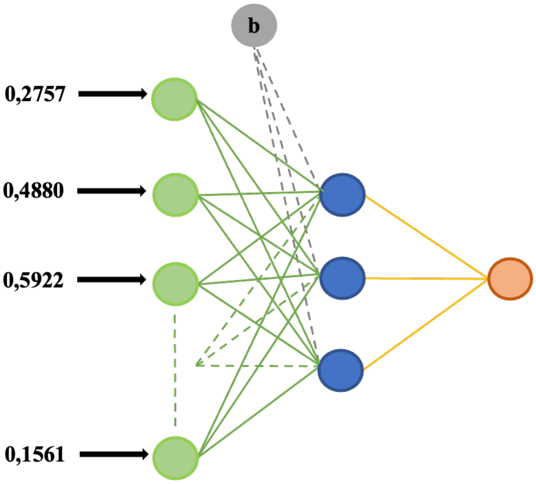


Figure 2: Simulation architecture model of the ANN method.

$v_3 = [0,3 \quad 0,1 \quad 0,2 \quad 0,3 \quad 0,1 \quad \ldots \quad 0,3]$ (weight of the edge leading to z_3)
$w_1 = 0,1; \quad w_2 = 0,2; \quad w_3 = 0,3$ (weight of the edge leading to $\hat{y}$)

(a) Feed-forward stage

The information from the input layer is sent to the hidden layer.

$z_{in} = x_i v + b$
$z_{in1,1} = x_1 v_{1,1} + b = 0,2757 \times 0,1 + 0,1 = 0,12757$
$z_{in1,2} = x_2 v_{1,2} + b = 0,4880 \times 0,2 + 0,1 = 0,1976$
$z_{in1,3} = x_3 v_{1,3} + b = 0,5922 \times 0,3 + 0,1 = 0,27766$
$\vdots$
$z_{in1,58} = x_{58} v_{1,58} + b = 0,1561 \times 0,1 + 0,1 = 0,11561$
$z_{in1} = [0,12757 \quad 0,1976 \quad 0,27766 \quad \ldots \quad 0,11561]$
$z_{in2,1} = x_1 v_{2,1} + b = 0,2757 \times 0,2 + 0,1 = 0,15514$
$z_{in2,2} = x_2 v_{2,2} + b = 0,4880 \times 0,3 + 0,1 = 0,2464$
$z_{in2,3} = x_3 v_{2,3} + b = 0,5922 \times 0,1 + 0,1 = 0,15922$
$\vdots$
$z_{in2,58} = x_{58} v_{2,58} + b = 0,1561 \times 0,2 + 0,1 = 0,13122$
$z_{in_2} = [0,15514 \quad 0,2464 \quad 0,15922 \quad \ldots \quad 0,13122$
$z_{in3,1} = x_1 v_{3,1} + b = 0,2757 \times 0,3 + 0,1 = 0,18271$
$z_{in3,2} = x_2 v_{3,2} + b = 0,4880 \times 0,1 + 0,1 = 0,1488$
$z_{in3,3} = x_3 v_{3,3} + b = 0,5922 \times 0,2 + 0,1 = 0,21844$
$\vdots$
$z_{in3,58} = x_{58} v_{3,58} + b = 0,1561 \times 0,3 + 0,1 = 0,14683$
$z_{in_3} = [0,18271 \quad 0,1488 \quad 0,21844 \quad \ldots \quad 0,14683]$

(b) Activation stage z_{in}

The results of sending information are then activated using the sigmoid function to get the value in the hidden layer.

$$z_1 = \frac{1}{1 + e^{(-z_{in_i})}}$$

$$z_1 = \frac{1}{1 + e^{(-[0{,}12757 \quad 0{,}1976 \quad 0{,}27766 \quad \ldots \quad 0{,}11561])}} = [0,5318 \quad 0,5492 \quad 0,5690 \quad \ldots \quad 0,5289]$$

$$z_2 = \frac{1}{1 + e^{(-[0{,}15514 \quad 0{,}2464 \quad 0{,}15922 \quad \ldots \quad 0{,}13122])}} = [0,5387 \quad 0,5613 \quad 0,5397 \quad \ldots \quad 0,5774]$$

$$z_3 = \frac{1}{1 + e^{(-[0{,}18271 \quad 0{,}1488 \quad 0{,}21844 \quad \ldots \quad 0{,}14683])}} = [0,5456 \quad 0,5371 \quad 0,5544 \quad \ldots \quad 0,5366]$$

(c) Output Fully-Connected Neural Networks
Information from the *hidden layer* is sent to the *output layer* using Equation 7.

$$y_{in_1} = z_{1,1}w_1 + z_{1,2}w_1 + z_{1,3}w_1 + \cdots + z_{1,58}w_1 + b$$

$$= (0,5318 \times 0,1) + (0,5492 \times 0,1) + (0,5690 \times 0,1) + \cdots + (0,5298 \times 0,1) + 0,1 = \ldots$$

$$y_{in_2} = z_{2,1}w_2 + z_{2,2}w_2 + z_{2,3}w_2 + \cdots + z_{2,58}w_2 + b$$

$$= (0,5387 \times 0,2) + (0,5613 \times 0,2) + (0,5397 \times 0,2) + \cdots + (0,5774 \times 0,2) + 0,1 = \ldots$$

$$y_{in_3} = z_{3,1}w_3 + z_{3,2}w_3 + z_{3,3}w_3 + \cdots + z_{3,58}w_3 + b$$

$$= (0,5456 \times 0,3) + (0,5371 \times 0,3) + (0,5544 \times 0,3) + \cdots + (0,5366 \times 0,3) + 0,1 = \ldots$$

$$y_{in} = y_{in_1} + y_{in_2} + y_{in_3} = \ldots$$

(d) Activation stage y_{in}
The results of sending information are then activated using the sigmoid function to get the value at the output layer.

$$\hat{y} = \frac{1}{1 + e^{(-y_{in})}} = \ldots$$

(e) Calculating the *error output* $(\hat{y})$

$$\Delta k = (t - \hat{y}) \times dlogsig(y_{in}, \hat{y}) = \ldots$$

(f) Calculating the *error backpropagation*

$$\Delta in_{j_1} = v_{1,1}\Delta k + v_{2,1}\Delta k + v_{3,1}\Delta k + \cdots + v_{58,1}\Delta k$$

$$\Delta in_{j_2} = v_{1,2}\Delta k + v_{2,2}\Delta k + v_{3,2}\Delta k + \cdots + v_{58,2}\Delta k = \ldots$$

$$\Delta in_{j_3} = v_{1,3}\Delta k + v_{2,3}\Delta k + v_{3,3}\Delta k + \cdots + v_{58,3}\Delta k = \ldots$$

(g) Activating *error backpropagation* with derivative log sigmoid

$$\Delta in_{j_1} = dlogsig(z_{in_1}, z_1) \times \Delta in_{j_1} = \ldots$$

$$\Delta in_{j_2} = dlogsig(z_{in_2}, z_2) \times \Delta in_{j_2} = \ldots$$

$$\Delta in_{j_3} = dlogsig(z_{in_3}, z_3) \times \Delta in_{j_3} = \ldots$$

(h) Updating weights and biases

$$v_1 = v_1 + \alpha \times \Delta k \times z_{in_1} = \ldots$$

$$v_2 = v_1 + \alpha \times \Delta k \times z_{in_1} = \ldots$$

$$v_3 = v_1 + \alpha \times \Delta k \times z_{in_1} = \ldots$$

Data from day 2 through day 4 were processed similarly from **Steps** $1 - 18$.

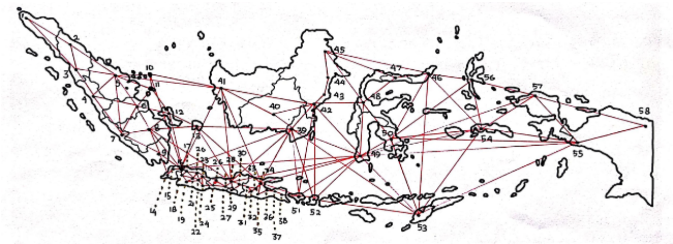


Figure 3: Map of Distribution of Branch Offices of Expedition Service Company throughout Indonesia.

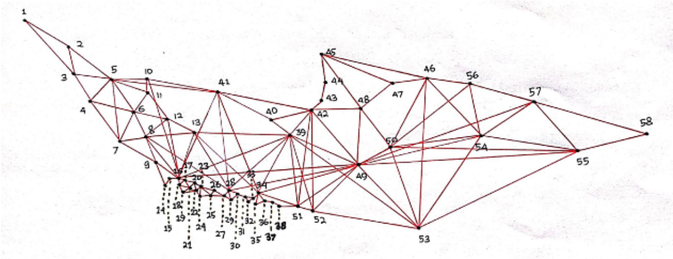


Figure 4: Spatial Graph Networks

3 Results and Discussion

3.1 Map of City Branch Office Locations

The locations of expedition service company branch offices throughout Indonesia are marked with black dots on the map of Indonesia, and each point is connected by a red line based on land, sea, and air transportation routes that connect each city location of the expedition service company branch office in real conditions.

1 = Aceh	16 = Jakarta	31 = Madiun	45 = Tarakan
2 = Medan	17 = Bekasi	32 = Kediri	46 = Manado
3 = Silangit	18 = Bogor	33 = Mojokerto	47 = Gorontalo
4 = Padang	19 = Sukabumi	34 = Surabaya	48 = Palu
5 = Pekanbaru	20 = Karawang	35 = Malang	49 = Ujung Pandang
6 = Jambi	21 = Purwakarta	36 = Pasuruan	50 = Kendari
7 = Bengkulu	22 = Bandung	37 = Probolinggo	51 = Denpasar
8 = Palembang	23 = Cirebon	38 = Jember	52 = Mataram
9 = Bandar Lampung	24 = Tasikmalaya	39 = Banjarmasin	53 = Kupang
10 = Batam	25 = Cilacap	40 = Palangkaraya	54 = Ambon
11 = Tanjung Pinang	26 = Tegal	41 = Pontianak	55 = Timika
12 = Pangkal Pinang	27 = Magelang	42 = Balikpapan	56 = Ternate
13 = Tanjung Pandan	28 = Semarang	43 = Samarinda	57 = Sorong
14 = Cilegon	29 = Yogyakarta	44 = Bontang	58 = Jayapura
15 = Tangerang	30 = Solo		

Based on the branch office distribution map in Figure 3, a spatial graph can be formed by taking only the vertices and connecting edges between the vertices so that the spatial graph is obtained as in Figure **??** The spatial graph will give its adjacency matrix, which will be used in the node embedding process using the GNN method.

3.2 Goods Distribution Data

The data used in this research is simulation data built randomly by considering the location of branch offices in a province and its islands. The data is compiled in Microsoft Excel, and the data is the number of goods that have been recorded at a branch office because the distribution is through a branch office that records both goods sent from the city, goods with the final destination of the city, and goods that only stop in the city. The data used in this study amounted to 26.100 data, which comes from $58(city) \times 15(feature) \times 30(days)$. An example of structuring simulation data in Microsoft Excel is presented in Table 1.

Table 1: Example of simulation data organization in Excel

H-1	Berkas	Makanan	Pakaian	Pecah Belah	Minuman	· · ·	Sepatu	Tas	Obat	H-2	· · ·	H-30
Banda Aceh	26	150	159	4	138	· · ·	55	18	59	· · ·	· · ·	· · ·
Medan	89	182	184	13	174	· · ·	57	18	78	· · ·	· · ·	· · ·
Silangit	46	74	114	5	68	· · ·	16	4	27	· · ·	· · ·	· · ·
Padang	82	179	112	25	142	· · ·	95	12	79	· · ·	· · ·	· · ·
Pekan Baru	41	44	90	9	42	· · ·	59	8	50	· · ·	· · ·	· · ·
Jambi	47	33	75	36	42	· · ·	32	7	58	· · ·	· · ·	· · ·
Bengkulu	54	72	70	21	83	· · ·	35	8	57	· · ·	· · ·	· · ·
Palembang	48	195	107	24	148	· · ·	30	21	49	· · ·	· · ·	· · ·
· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·
Ternate	33	17	40	18	45	· · ·	18	3	37	· · ·	· · ·	· · ·
Sorong	38	43	169	26	51	· · ·	27	10	49	· · ·	· · ·	· · ·
Jayapura	12	39	85	11	46	· · ·	26	4	54	· · ·	· · ·	· · ·

Table 2: Adjacency matrix of spatial graphs

A	1	2	3	4	5	6	7	8	9	10	11	· · ·	55	56	57	58
1	0	1	1	0	0	0	0	0	0	0	0	· · ·	0	0	0	0
2	1	0	1	0	1	0	0	0	0	0	0	· · ·	0	0	0	0
3	1	1	0	1	1	0	0	0	0	0	0	· · ·	0	0	0	0
4	0	0	1	0	1	1	1	0	0	0	0	· · ·	0	0	0	0
5	0	1	1	1	0	1	1	0	0	1	1	· · ·	0	0	0	0
6	0	0	0	1	1	0	1	1	0	0	1	· · ·	0	0	0	0
7	0	0	0	1	1	1	0	1	1	0	0	· · ·	0	0	0	0
8	0	0	0	0	0	1	1	0	1	0	0	· · ·	0	0	0	0
9	0	0	0	0	0	0	1	1	0	0	0	· · ·	0	0	0	0
10	0	0	0	0	1	0	0	0	0	0	1	· · ·	0	0	0	0
11	0	0	0	0	1	1	0	0	0	1	0	· · ·	0	0	0	0
· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·
55	0	0	0	0	0	0	0	0	0	0	0	· · ·	0	0	1	1
56	0	0	0	0	0	0	0	0	0	0	0	· · ·	0	0	1	0
57	0	0	0	0	0	0	0	0	0	0	0	· · ·	1	1	0	1
58	0	0	0	0	0	0	0	0	0	0	0	· · ·	1	0	1	0

3.3 Adjacency Matrix

Based on the spatial graph formed in Figure 4, the neighborhood relationship of each point in the spatial graph is converted into an adjacency matrix based on the rules in Equation 3, then presented in Table 2.

3.4 ANN Method

The ANN model in this study was run with Matlab R2018b programming software; see the Appendix. The program script is written in the editor, and there are three programs made in this study, namely (1) a program for node embeddings of the features of each point, (2) a program for ANN training, and (3) a program for ANN testing and prediction. The first is a node embedding program with ten iterations of the GNN method that goes through the message passing, aggregate, and update stages. The distribution data of goods delivery from day 1 to day two is entered one by one (58 city vertex with 15 features) into the program to obtain the node embeddings. It is entered one by one day (58 city vertices with 15 features) into this program to get the node embedding features per vertex. The resulting vector from each day is stored. The first program produces output as in Figures 5 and 6; the first is a plot of the vertex embeddings results from each point with the three highest vertex embeddings values consecutively 49 vertex, 39 vertex, and 16 vertex.

The second output is each iteration's loss (error) history, as shown in Figure 6, from the first iteration to the tenth iteration. The loss value is getting smaller (closer to zero), so the more iterations performed, the smaller the loss value will be, or the closer the distance between features in the embedding space. The number of iterations will also affect the value of node embeddings; the more iterations, the value of each point's node embeddings will also be smaller.

The resulting graph embedding from day 1 to day 30 is called by the second program to be trained using different architectures with different numbers of hidden layers and nodes per hidden layer. We use four ANN models available in Matlab, namely feed-forward network, fit network, pattern network, and cascade forward network. The ANN architecture is 4-6-7, 4-41-1, and 4-6-6; see Figure 7. The data is trained individually regarding the four different architectures and hidden layer arrangement on the data from days 1-20. The results of the training process are shown in Figure 8. The data that has been trained will be tested using the third program, with the data tested in the model from day 21 to day 30. Again, the data was tested one by one using the four architectures of ANN models in Figure 7. The testing process results are the graphs shown in Figure 9.

The second output of the third program is a prediction graph, as shown in Figure 10, which is a purple graph. The graph results from predicting the pile of goods from each vertex (city) on day 31. The highest result is at vertex 49 of Ujung Pandang City, the second is at vertex 39 of Samarinda City, and the third is at vertex 16 of Jakarta City. The largest pile of goods in

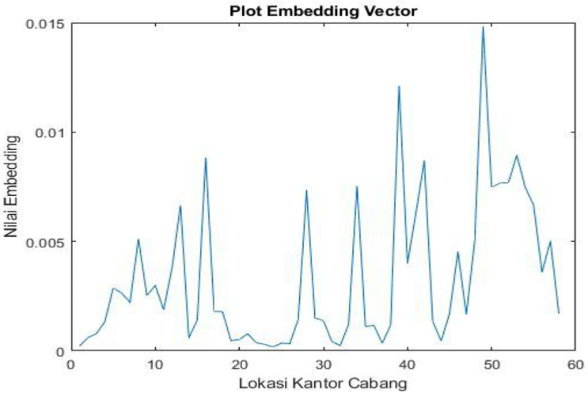


Figure 5: Results of Node Embeddings of Each Vertex.

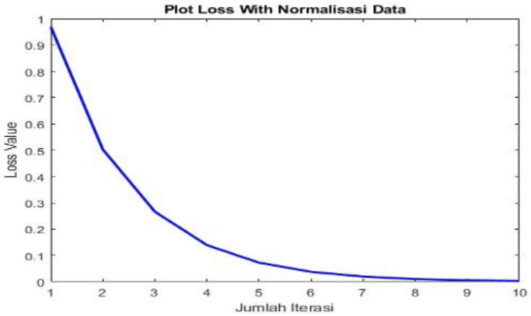
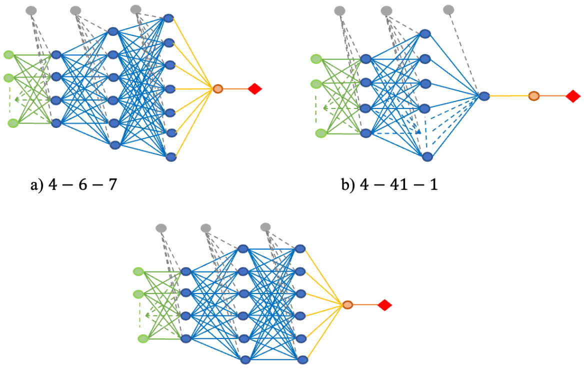


Figure 6: Results of Node Embeddings of Each Vertex.

the three cities will experience excess distribution capacity of goods delivery on day 31, so it will experience delays in the distribution process of shipping goods to the surrounding cities.

The results of all the performance experiments using several different architectures and hidden layer arrangements are in Table 3.

Based on the results obtained in Table 3, the smallest to highest MSE of the models successively is obtained by the cascade forward network, fit network, feed-forward network, and pattern network architectures. Based on the training time from the fastest to the longest model are fit networks, pattern networks, feed-forward networks, and cascade forward networks. Based on the regression value, from the largest to the smallest, we have a cascade forward network, fit network, and feed-forward network. In contrast, the pattern network architecture has no known regression value, so it is said to be invalid.


a) 4 − 6 − 7

b) 4 − 41 − 1

c) 4 − 6 − 6

Figure 7: Number of hidden layers and nodes per hidden layer in the study.

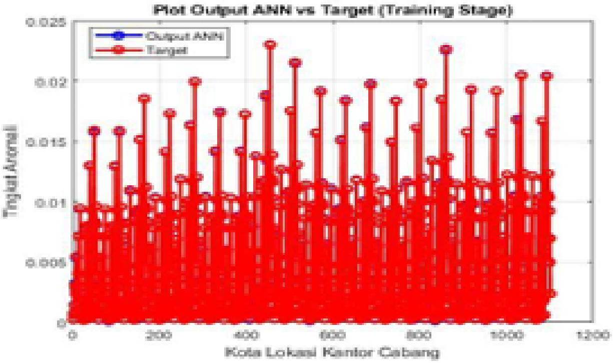


Figure 8: Number of hidden layers and nodes per hidden layer in the study.

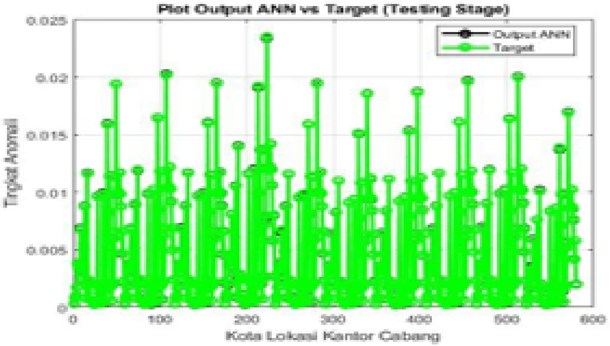


Figure 9: Number of hidden layers and nodes per hidden layer in the study.

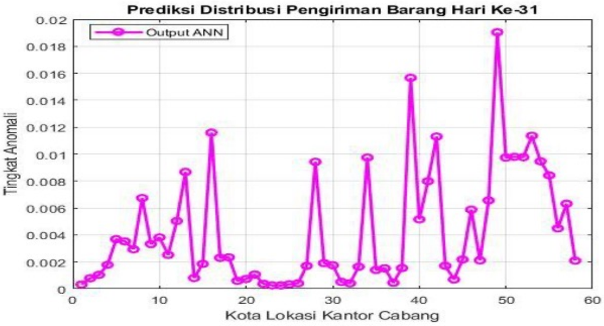


Figure 10: Number of hidden layers and nodes per hidden layer in the study.

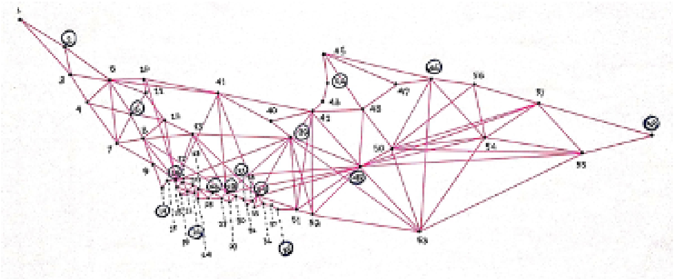


Figure 11: Dominating set of spatial graph.

Table 3: The performance of GNN models in terms of MSE, training times, and regression

Architecture	Hidden Layer	MSE (Training)	MSE (Testing)	Iterations	Training Time (s)	Performance	Regression
Feed-forward	4-6-7	1.4145×10^{-8}	1.3153×10^{-8}	22	1	1.4145×10^{-8}	0.99963
	4-41-1	8.8785×10^{-9}	7.7193×10^{-9}	17	10	8.8785×10^{-9}	0.99976
	4-6-6	1.165×10^{-8}	1.0545×10^{-8}	18	0	1.165×10^{-8}	0.99969
Fit network	4-6-7	1.3861×10^{-8}	1.1088×10^{-8}	14	0	1.3861×10^{-8}	0.99963
	4-41-1	7.7548×10^{-9}	6.5516×10^{-9}	17	2	7.7548×10^{-9}	0.99979
	4-6-6	1.1062×10^{-8}	1.0571×10^{-8}	39	1	1.1062×10^{-8}	0.99971
Pattern network	4-6-7	2.3601×10^{-8}	2.2618×10^{-8}	14	1	2.3601×10^{-8}	0.99881
	4-41-1	1.1537×10^{-8}	1.3373×10^{-8}	20	2	1.1537×10^{-8}	0.9989
	4-6-6	3.4455×10^{-8}	3.244×10^{-8}	17	1	3.4455×10^{-8}	0.99872
Cascade forward	4-6-7	2.9308×10^{-9}	3.268×10^{-9}	27	28	2.9308×10^{-9}	0.99992
	4-41-1	1.6714×10^{-9}	1.8592×10^{-9}	14	72	$\mathbf{1.6714 \times 10^{-9}}$	0.99996
	4-6-6	4.4037×10^{-9}	4.2242×10^{-9}	14	7	4.4037×10^{-9}	0.99988

3.5 Dominating Set

The dominating set formed from the spatial graph in Figure 4 is $S = 2,\ 6,\ 14,\ 16,\ 22,\ 26,\ 28,\ 31,\ 34,\ 38,\ 39,\ 44,\ 46,\ 49,\ 58$ which is located at 15 city points as shown in Figure 11. They are as follows.

Based on this dominating set, the set of vertices dominating the vertices around them in the spatial graph indicates that the vertices in experience excess capacity in distributing goods. This is because the points in are adjacent to several points in the spatial graph, which causes the process of distributing goods from cities around to pass through several points in . Distributing goods through a city in S can either be aimed at that city or only stop in that city and then distributed to adjacent cities. This results in a buildup of goods at the branch offices of the shipping company, with the highest anomaly occurring at the vertex of number 49, namely Ujung Pandang city, followed by the vertex of number 39, Banjarmasin city, and the third is the vertex of number 16, namely Jakarta city. The result of this dominating set is in line with the prediction of branch office locations that experience excess capacity in distributing goods using the ANN method, as shown in Figure 10.

2. Medan	26. Tegal	39. Banjarmasin
6. Jambi	28. Semarang	44. Bontang
14. Cilegon	31. Madiun	46. Manado
16. Jakarta	34. Surabaya	49. Ujung Pandang
22. Bandung	38. Jember	58. Jayapura

Based on this dominating set, the set of vertices dominating the vertices around them in the spatial graph indicates that the vertices in S experience excess capacity in distributing goods. This is because the points in S are adjacent to several points in the spatial graph, which causes the process of distributing goods from cities around S to pass through several points in S.

Distributing goods through a city in S can either be aimed at that city or only stop in that city and then distributed to adjacent cities. This results in a buildup of goods at the branch offices of the shipping company, with the highest anomaly occurring at the vertex of number 49, namely Ujung Pandang city, followed by the vertex of number 39, Banjarmasin city, and the third is the vertex of number 16, namely Jakarta city. The result of this dominating set is in line with the prediction of branch office locations that experience excess capacity in distributing goods using the ANN method, as shown in Figure 10.

4 Discussion

The result of the GNN method in this study is a matrix of feature embeddings of spatial vertices in the distribution of goods delivery through an expedition service company in Indonesia. The embedding process can reduce the dimensionality of vertex features from the original fifteen to one-dimensional features through the message passing, aggregate, and update stages. Each point stores information or messages that will be passed on to other points during the message-passing stage by considering the spatial graph's loop adjacency matrix $B = A + I$, where A is the adjacency matrix, I is the diagonal matrix, and B is the adjacency matrix, including their self adjacency. Therefore, in the message passing stage, matrix B refers to adding messages in the aggregate process after they are sent. The proximity of two vertices after the aggregate stage is measured using an error value as a benchmark produced using weights of 0.1 and 10 iterations. After obtaining the embedding matrix, the location of a branch office that experiences excess capacity in distributing goods will be predicted using the ANN model. The ANN model consists of training, testing, and forecasting (prediction) using four models and three ANN architectures used during the training stage. Then, the best model and architecture will be used in the testing stage. The mean square error (MSE) value is used as a benchmark in determining the best model and architecture. Based on the testing results, ANN's best model and architecture are cascade forward network$4 - 41 - 1$. This result is in line with the study conducted by Alrubaie (2021), which showed that the cascade forward network model could simulate the VIE solution value with a 100% percentage. The best model and architecture of ANN are used to predict the location of branch offices of the expedition service company that experience excess capacity in distributing goods on the 31st day. The prediction result is shown in Figure 10.

5 Conclusion

The best model architecture is the cascade forward network of 4-41-1 architecture, since it has the smallest MSE value and the highest regression value compared to other model architectures. However, the fit network architecture is the fastest in the training process compared to other model architectures. However, the cascade forward network architecture gains better MSE and regression values. Therefore, if you need the best results, you can use the cascade forward network architecture, but it has a longer training time.

The prediction results show that three cities experience excess distribution capacity with the highest stack of goods, namely cities 49, 39, and 16. This is in line with the dominating set formed from the spatial graph, and these three cities are included as members of the set of points in the dominating set. Therefore, the logistics company needs to evaluate and make improvements according to the needs and capabilities of the company. Based on the results of this study, the STGNN method combined with dominating set techniques can solve the problem of excess distribution capacity of goods delivery through a logistics company effectively.

Acknowledgment

We gratefully acknowledge PUI-PT Combinatoric and Graph, CGANT-Universitas Jember, for providing supervision, suggestions and collaborations to finishing this paper. We also would like to thank to LP2M-Universitas Jember for the research support of the year 2024.

Competing Interests

The authors declare that they have no competing interests.

Authors Contributions

BA and IHA designed the study stages, IHA analyzed the dominating set, D developed the algorithm and programming of graph embedding, and RIB developed the algorithm and programming for times series forecasting using ANN, which includes training, testing, and forecasting process. BA and IHA experimented on this problem. MF analyzed the performance of the models. All authors have contributed to finishing the manuscript.

References

[1] URL. (Accessed December 2, 2022). [Online]. Available: https://kargo.tech/blog/daftar-jasa-kurir-pengiriman-barang-yang-mana-pilihanmu/

[2] ——. (Accessed November 7, 2022). [Online]. Available: https://kargo.tech/blog/5-jasa-ekspedisi-yang-melayani-pengiriman-barang-besar-2/

[3] Eko. (2022) Jne trucking, more than 20 days and the package has not been received yet. (Accessed November 7, 2022). [Online]. Available: https://www.google.com/amp/s/mediakonsumen.com/2022/10/17/surat-pembaca/jne-trucking-lebih-dari-20-hari-paket-masih-belum-diterima/amp

[4] R. Kumalasari. Everything you need to know about shipping goods!

[5] D. Harliyuni, Dafik, Slamin, Z. R. Ridlo, and R. Alfarisi, "On the spatial graph neural network analysis together with local vertex irregular reflexive coloring for time series forecasting on passenger density at bus station," *Advances in Intelligent System Research*, vol. 177, pp. 305–323, 2023.

[6] S. A. H. Alrubaie, "Cascade-forward neural network for volterra integral equation solution," *Ibn Al Haitham Journal for Pure and Applied Science*, vol. 34, pp. 104–114, 2021.

[7] L. Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*. Pearson, 1993.

[8] S. Setti, I. A. R. Simbolon, M. Syafiq, and I. Parlina, "The application of artificial neural network in predicting the amount of crude oil exports in indonesia," *Journal Of Informatics And Telecommunication Engineering*, vol. 2, pp. 31–38, 2018.

[9] A. Gedik, "Short-term traffic volume prediction for the merging roads by artificial neural network," *El-Cezeri Journal of Science and Engineering*, vol. 7, no. 3, pp. 1496–1508, 2020.

[10] P. Yilmaz, S. Akcakaya, S. Ozkaya, and A. Cetin, "Machine learning based music genre classification and recommendation system," *El-Cezeri Journal of Science and Engineering*, vol. 9, no. 4, pp. 1560–1571, 2022.

[11] M. R. T. Dale, *Applying Graph Theory in Ecological Research*. TJ International Ltd, 2017.

[12] Kiki and S. Kusumadewi, "Artificial neural network with backpropagation method for detecting psychological disorders," *Media Informatika*, vol. 2, pp. 1–11, 2004.

[13] W. L. Hamilton, *Graph Representation Learning*. Morgan & Claypool Publisher, 2020.

[14] A. Muklisin, I. M. Tirta, Dafik, R. I. Baihaki, and A. I. Kristiana, "The analysis of airport flow by using spatial-temporal graph neural networks and resolving efficient dominating set," *Advances in Intelligent System Research*, vol. 177, pp. 3–20, 2023.

[15] A. Singh, "Classification of malware in https traffic using machine learning approach," *El-Cezeri Journal of Science and Engineering*, vol. 9, no. 2, pp. 644–655, 2022.

[16] G. Chartrand and P. Zhang, *A First Course in Graph Theory*. Dover Publications, 2012.

[17] J. G. Nicholls, A. R. Martin, P. A. Fuchs, D. A. Brown, M. E. Diamond, and D. A. Weisblat, *From Neuron to Brain, Fifth Edition*. Sinauer Associates, Inc., 2012.

[18] M. Aziza, Dafik, A. Kristiana, R. Alfarisi, and D. Wardani, "On resolving perfect dominating number of comb product of special graphs," *Journal of Physics: Conference Series*, vol. 1832, 2021.

[19] Dafik, I. H. Agustin, and A. R. Wardani, "The number of locating independent dominating set on generalized corona product graphs," *Advances in Mathematics Scientific Journal*, vol. 9, pp. 4873–4891, 2020.

[20] D. A. R. Wardani, Dafik, and I. H. Agustin, "The locating dominating set (lds) of generalized of corona product of path graph and any graphs," *Journal of Physics: Conference Series*, vol. 1465, 2020.

[21] N. Mauliska, W. Lestari, E. T. Wisudaningsih, and M. H. Islam, "Application of spatial-temporal graph neural networks for forecasting data time series river pollution waste content in probolinggo," *Advances in Intelligent System Research*, vol. 177, pp. 257–272, 2023.