Application of Spatial Temporal Graph Neural Networks for Forecasting Data Time Series River Pollution Waste Content in Probolinggo

Nur Mauliska[1], Wahyu Lestari[1](✉), Endah Tri Wisudaningsih[2], and Muhammad Hifdil Islam[2]

[1] Department of Mathematics Education, Zainul Hasan Genggong Islamic University, Probolinggo, Indonesia
nurmauliska4@gmail.com, why.lestari94@gmail.com
[2] Departement of Islamic Religious Education, Zainul Hasan Genggong Islamic University, Probolinggo, Indonesia

Abstract. Flooding has become a serious problem in Probolinggo. One of the causes of flooding is the accumulation of garbage in the river. Garbage can also cause river water pollution. To measure water pollution, we use a pH meter. SRAC (Strong Rainbow Antimagic Coloring) is the smallest number of colors taken from all rainbow colorings and is induced by strong antimagic rainbow labeling from G. The coloring resulting from SRAC determine the placement of river waste cleaner. In this study we apply spatial temporal graph neural networks (STGNN) to predict river waste in the future. In this study, the best error value is 3.2262×10^{-6} which is generated using a weight of 0.1 and 12 iterations. Based on the test results, the smalest MSE 5.0687×10^{-9}, obtained from ANN model Cascadeforwardnet with 468 architectures.

Keywords: SRAC · STGNN · Artificial Neural Networks · River Waste Time Series Analysis

1 Introduction

Most of the watersheds in Indonesia have been damaged, there are as many as 62 of the 82 major rivers classified as crisis rivers, including the watersheds of Probolinggo Regency. Most of the damage to the river is caused by various kinds of human activities which require the river to become a free garbage and waste disposal site [12]. All forms of waste are discharged into rivers without going through any processing. The water quality of the Pekalen River in Probolinggo Regency is polluted by waste from households, industry, fisheries, and agriculture [10]. The impact is very dangerous, because river water is still used for daily needs, be it bathing, washing, or drinking water [1]. Garbage can also cause river water pollution, thus threatening fish habitat in the river.

The problems of the Pekalen River are becoming very complex in line with the high activity of the people around the rivers of the Pekalen watershed. Starting from

I. H. Agustin (Ed.): ICONNSMAL 2022, AISR 177, pp. 257–272, 2023.
https://doi.org/10.2991/978-94-6463-174-6_19

the occurrence of river water pollution, narrowing of the river body, high erosion and sedimentation, which leads to frequent flooding in the Pekalen River basin. The efforts to overcome this problem are by cleaning the Pekalen River regular basis. River cleaning can be determined by the graph concept in mathematics, namely Strong Rainbow Antimagic Coloring.

SRAC (Strong Rainbow Antimagic Coloring) is the smallest number of colors taken from all the rainbow dyes and induced by the strong rainbow antimagic label of G. The resulting staining from SRAC determine the placement of river waste cleaner. A path in P is called a rainbow path if no two edges in P are the same color [9]. A graph G is called rainbow connected with coloring c if for every $(u, v) \in G$ there is a rainbow path from u to v. If there are k colors in G then c is the k rainbow coloring. The rainbow connection number of a connected graph is denoted by $rc(G)$, which is the minimum number of colors needed to make a rainbow connected graph G [3]. Strong rainbow k-coloring is c coloring if for every vertex u and v there is a rainbow path with the same length as the distance u and v.

Deep Learning is a branch of machine learning based on Artificial Neural Networks (ANN). Artificial Neural Network is a computational method that mimics a biological neural network system using basic non-linear computational elements called neurons [5]. Neurons are organized as an interconnected network [8], making it similar to a human neural network. The arrangement of neurons in layers and the pattern of connections within and between layers is called network architecture. The architecture of a neural network consists of an input layer, a hidden layer and an output layer [6]. ANN was formed to solve certain problems such as pattern recognition or classification due to the learning process. ANNs are developed and used for image recognition, natural language processing, soil moisture forecasting and so on [4].

Graph Neural Network is a type of Neural Network that directly operates on graph data structures. GNN is one of the deep learning architectures used to solve machine learning problems on graphical data [11]. Basically, every node in a graph is associated with a label. STGNN is a type of GNN with a spatial and temporal dependence model used in data analysis with data collected in space and time [7]. It describes phenomena at a specific location and time. For example, forecasting of river pollution levels in the future.

In this study we apply Spatial Temporal Graph Neural Networks (STGNN) to predict future river waste, and apply the Strong Rainbow Antimagic Coloring (SRAC) Concept to determine the placement of waste cleaners in the Pekalen Watershed, Probolinggo Regency.

2 Methods

The method we used in this research is analytic and experimental methods. This research was conducted for fourteen days, with three types of feature data, sub-watershed broad, water discharge, and water pH. We used a mathematical deductive approach in analytical studies to illustrate findings. In the experimental method, we used computer programming to carry out numerical simulations. In this research, we will analyze river waste anomalies from eight rivers in Probolinggo Regency, East Java, Indonesia. The process divided into three steps are: 1) from a graph with three data features the process of inserting single layer GNN nodes will be displayed; 2) developed with the STGNN programming using 70% of the input data obtained from the node insertion process to train the model, test, then predict river waste anomalies.

The algorithm we used to study river waste anomalies using STGNN combined with strong rainbow antimagic coloring is shown below. The results of graph coloring using the strong rainbow antimagic coloring method are used to determine the placement of river waste cleaner. Strong rainbow antimagic coloring is the minimum number of colors taken from all strong rainbow colorings induced by strong rainbow antimagic labeling from G.

2.1 Single Layer GNN Algorithm

River waste flow can be predicted by using STGNN combined with strong rainbow antimagic coloring. The following is a single layer GNN algorithm [2].

- Step 0. Given that a graph $G(V, E)$ of order n and feature matrix $H_{n\times m}$ of n vertices and m features, and give a tolerance ϵ.
- Step 1. Determine the matrix adjacency A of graph G and set a matrix $B = A + I$, where I is an identity matrix.
- Step 2. Initialize weights W, bias β, learning rate α. (For simplicity, set $W_{m\times 1} = [w_1 w_2 \dots w_m]$, where $0 < w_j < 1$, bias $\beta = 0$ and $0 < \alpha < 1$).
- Step 3. Multiply weight matrix with vertex features, by setting a message function ${m_u}^l = MSG^l(h_u^{l-1})$, for linear layer ${m_u}^l = W^l(h_u^{l-1})$.
- Step 4. Aggregate the messages from vertex v's neighbors, by setting function ${h_v}^l = AGG^l\{h_u^{l-1}, u \in N(v)\}$, and by applying the **sum**$(\cdot)$ function ${h_v}^l = SUM^l\{h_u^{l-1}, u \in N(v)\}$ in regards with matrix B.
- Step 5. Determine the error, by setting $error^l = \frac{|\left|h_{v_i} - h_{v_j}\right||2}{|E|}$, where v_i, v_j are any two adjacent vertices.
- Step 6. Observe whether $error \leq \epsilon$ or not. If **yes** then stop, if **not** then do Step 7 to update the learning weight matrix W.
- Step 7. Update the learning weight matrix by setting $W^{l+1} = W_j^l + \alpha \times z_j \times e^l$ where z_j is the sum of each column in the $H_{v_i}^l$ and divide by the number of nodes.
- Step 8. Do Step 3–6 till the $error \leq \epsilon$.
- Step 9. Save the embedding results into a vector, by naming the vector file with *embedding_data.mat*. When the data is a time series data, then do the same proses for the next time data observation.
- Step 10. Load the *embedding_data.mat* then use the time series machine learning to do forecasting.
- Step 11. Have the best training, testing and forecasting results, then STOP.

3 Research Findings

Here we will explain the results of the research we have done as follows. In the first stage, we will show the process of inserting analytic point features and a strong rainbow antimagic coloring graph from the Pekalen River, Probolinggo Regency. In the second

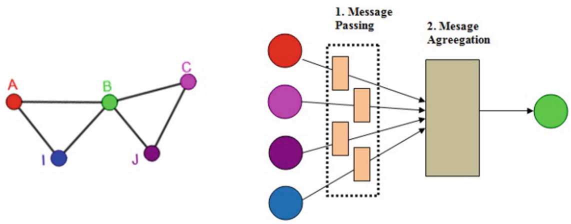

Fig.1. Illustration of nodes embedding on Graph Neural Networks

stage, we used observed river waste data to obtain the STGNN model, so that with the obtained STGNN model, time series forecasting of river waste anomalies can be carried out. Figure 1 shows the illustration of nodes embedding on Graph Neural Networks.

3.1 The Example for Illustration

To have more understanding about the graph neural network, let us give some technical examples about a specific graph with a specific feature of each vertex/node.

Example 1. *Given that a graph G of order five. Suppose that vertex and edge sets are $V(G) = \{v_1, v_2, v_3, v_4, v_5\}$ and $V(G) = \{v_1v_2, v_1v_3, v_2v_3, v_2v_4, v_2v_5, v_4v_5\}$, respectively. Given that feature node as.*

$$H_{v_i}^0 = \begin{bmatrix} 0,30 & 0,90 & 0,43 \\ 0,55 & 0,80 & 0,39 \\ 0,17 & 0,48 & 0,48 \\ 0,25 & 0,46 & 0,24 \\ 0,33 & 0,36 & 0,10 \end{bmatrix}$$

Obtain the nodes embedding with one hidden layer with one neuron, and with minimal loss function.

Solution. By the above graph, we can determine the adjacency, identity, and loop-adjacency matrices as follows.

$$A(G) = \begin{bmatrix} 0 & 1 & 1\,0\,0 \\ 1 & 0 & 1\,0\,0 \\ 1 & 1 & 0\,1\,1 \\ 0 & 0 & 1\,0\,1 \\ 0 & 0 & 1\,1\,0 \end{bmatrix}, I = \begin{bmatrix} 1 & 0 & 0\,0\,0 \\ 0 & 1 & 0\,0\,0 \\ 0 & 0 & 1\,0\,0 \\ 0 & 0 & 0\,1\,0 \\ 0 & 0 & 0\,0\,1 \end{bmatrix},$$

$$B = A + I = \begin{bmatrix} 1 & 1 & 1\,0\,0 \\ 1 & 1 & 1\,0\,0 \\ 1 & 1 & 1\,1\,1 \\ 0 & 0 & 1\,1\,1 \\ 0 & 0 & 1\,1\,1 \end{bmatrix}$$

We can start the technical calculation by initiating the learning weight $W = [0.20.20.2]$ of $(1,3)$-matrix. The first iteration can be described as follows:

$$\begin{aligned}
m^{l}_{v_i} &= W^{l}.H^{l-1}_{v_i}, \text{ where } i = 1,2,3,4,5\\
m^{1}_{v_1} &= W^{1}.H^{0}_{v_1}\\
&= [0.20.20.2].[0.300.900.43] = [0.326]\\
m^{1}_{v_2} &= W^{1}.H^{0}_{v_2}\\
&= [0.20.20.2].[0.550.800.39] = [0.348]\\
m^{1}_{v_3} &= W^{1}.H^{0}_{v_3}\\
&= [0.20.20.2].[0.170.480.48] = [0.226]\\
m^{1}_{v_4} &= W^{1}.H^{0}_{v_4}\\
&= [0.20.20.2].[0.250.460.24] = [0.190]\\
m^{1}_{v_5} &= W^{1}.H^{0}_{v_5}\\
&= [0.20.20.2].[0.330.360.10] = [0.158]
\end{aligned}$$

thus, we have $m^1_{v_i}$: $m^1_{v_i} = \begin{bmatrix} 0.326\\ 0.348\\ 0.226\\ 0.190\\ 0.158 \end{bmatrix}$

By considering the matrix B, we only include the non zeros element of $m^1_{v_i}$, thus we have.

$$m^1_{v_1} = \begin{bmatrix} 0.326\\ 0.348\\ 0.226 \end{bmatrix}, m^1_{v_2} = \begin{bmatrix} 0.326\\ 0.348\\ 0.226 \end{bmatrix}, m^1_{v_3} = \begin{bmatrix} 0.326\\ 0.348\\ 0.226\\ 0.190\\ 0.158 \end{bmatrix},$$

$$m^1_{v_4} = \begin{bmatrix} 0.226\\ 0.190\\ 0.158 \end{bmatrix}, m^1_{v_5} = \begin{bmatrix} 0.226\\ 0.190\\ 0.158 \end{bmatrix}$$

Take the sum of the elements of each nodes embedding are as follows: $h^1_{v_1} = 0.9$, $h^1_{v_2} = 0.9$, $h^1_{v_3} = 1.248$, $h^1_{v_4} = 0.574$, $h^1_{v_5} = 0.574$. Thus, we have the first iteration of aggregation $h^1_{v_i} = \begin{bmatrix} 0.9\\ 0.9\\ 1.248\\ 0.574\\ 0.574 \end{bmatrix}$ where $i = 1,2,3,4,5$.

The loss (e) can be calculated as.

$$e^1 = \frac{\left|\left|h_{v_i}^1 - h_{v_j}^1\right|\right| inf}{|E(G)|}, \text{ where} i, j \in \{1, 2, 3, 4, 5\} = 0.1087$$

In the second iteration, we need update $H_{v_i}^{l-1}$ first:

$$H_{v_i}^{l-1} = \frac{H_{v_i}^{l-2}}{\sum(H_{v_i}^{l-2})} \times h_{v_i}^{l-1} \text{where, } i = 1, 2, 3, 4, 5$$

$$H_{v_1}^1 = \frac{\left[0.30\ 0.90\ 0.43\right]}{\sum\left[0.30\ 0.90\ 0.43\right]} \times 0.9 = \left[0.166\ 0.497\ 0.237\right]$$

$$H_{v_2}^1 = \frac{\left[0.55\ 0.80\ 0.39\right]}{\sum\left[0.55\ 0.80\ 0.39\right]} \times 0.9 = \left[0.284\ 0.414\ 0.212\right]$$

$$H_{v_3}^1 = \frac{\left[0.17\ 0.48\ 0.48\right]}{\sum\left[0.17\ 0.48\ 0.48\right]} \times 1.248 = \left[0.188\ 0.530\ 0.530\right]$$

$$H_{v_4}^1 = \frac{\left[0.25\ 0.46\ 0.24\right]}{\sum\left[0.25\ 0.46\ 0.24\right]} \times 0.574 = \left[0.151\ 0.278\ 0.145\right]$$

$$H_{v_5}^1 = \frac{\left[0.33\ 0.36\ 0.10\right]}{\sum\left[0.33\ 0.36\ 0.10\right]} \times 0.574 = \left[0.240\ 0.262\ 0.073\right]$$

thus, we have $H_{v_i}^1$:

$$H_{v_i}^1 = \begin{bmatrix} H_{v_1}^1 \\ H_{v_2}^1 \\ H_{v_3}^1 \\ H_{v_4}^1 \\ H_{v_5}^1 \end{bmatrix} = \begin{bmatrix} 0.166 & 0.497 & 0.237 \\ 0.284 & 0.414 & 0.212 \\ 0.188 & 0.530 & 0.530 \\ 0.151 & 0.278 & 0.145 \\ 0.240 & 0.262 & 0.073 \end{bmatrix}$$

Now, we need to update the learning weight. Given that the learning rate α, we can update the weight w as.

$$W^{k+1} = W_j^k + \alpha \times z_k \times e^k \text{where} j = 1, 2, 3$$

$$\begin{aligned} W^2 &= W_j^1 + \alpha \times z_k \times e^k, \text{ for } k = 1 \text{ and } \alpha = 0.1 \\ W_1^2 &= W_1^1 + \alpha \times z_1 \times e^1 \\ &= 0.2 + 0.1 \times 0.2058 \times 0.1087 = 0.2022 \\ W_2^2 &= W_2^1 + \alpha \times z_2 \times e^1 \\ &= 0.2 + 0.1 \times 0.3962 \times 0.1087 = 0.2043 \\ W_3^2 &= W_3^1 + \alpha \times z_3 \times e^1 \end{aligned}$$

$$= 0.2 + 0.1 \times 0.2394 \times 0.1087 = 0.2027$$

Thus, we have W^2:

$$W^2 = \left[W_1^2 W_2^2 W_3^2\right] = [0.20220.20430.2027]$$

By this new W^2 in hand, we can calculate the second iteration as follows.

$$\begin{aligned}
m_{v_i}^l &= W^l.H_{v_i}^{l-1} \text{where } i = 1, 2, 3, 4, 5, \\
m_{v_1}^2 &= W^2.H_{v_1}^1 \\
&= [0.20220.20430.2027].[0.1660.4970.237] \\
&= [0.1831] \\
m_{v_2}^2 &= W^2.H_{v_2}^1 \\
&= [0.20220.20430.2027].[0.2840.4140.212] \\
&= [0.1849] \\
m_{v_3}^2 &= W^2.H_{v_3}^1 \\
&= [0.20220.20430.2027].[0.1880.5300.530] \\
&= [0.2538] \\
m_{v_4}^2 &= W^2.H_{v_4}^1 \\
&= [0.20220.20430.2027].[0.1510.2780.145] \\
&= [0.1167] \\
m_{v_5}^2 &= W^2.H_{v_5}^1 \\
&= [0.20220.20430.2027].[0.2400.2620.073] \\
&= [0.1169]
\end{aligned}$$

thus, we have $m_{v_i}^2 = \begin{bmatrix} 0.1831 \\ 0.1849 \\ 0.2538 \\ 0.1167 \\ 0.1169 \end{bmatrix}$

By considering the matrix B, we only include the non zeros element of $h_{v_i}^2$, thus we have

$$m_{v_1}^2 = \begin{bmatrix} 0.1831 \\ 0.1849 \\ 0.2538 \end{bmatrix}, m_{v_2}^2 = \begin{bmatrix} 0.1831 \\ 0.1849 \\ 0.2538 \end{bmatrix},$$

$$m_{v_3}^2 = \begin{bmatrix} 0.1831 \\ 0.1849 \\ 0.2538 \\ 0.1167 \\ 0.1169 \end{bmatrix}, m_{v_4}^2 = \begin{bmatrix} 0.2538 \\ 0.1167 \\ 0.1169 \end{bmatrix}, m_{v_5}^2 = \begin{bmatrix} 0.2538 \\ 0.1167 \\ 0.1169 \end{bmatrix}$$

Take the sum of the elements of each nodes embedding are as follows: $h^2_{v_1} = 0.6218, h^2_{v_2} = 0.6218, h^2_{v_3} = 0.8554, h^2_{v_4} = 0.4874, h^2_{v_5} = 0.4874$. Thus, we have the second iteration of aggregation.

$$h^2_{v_i} = \begin{bmatrix} 0.6218 \\ 0.6218 \\ 0.8554 \\ 0.4874 \\ 0.4874 \end{bmatrix} \text{where } i = 1, 2, 3, 4, 5$$

The loss (e) can be calculated as

$$e^2 = \frac{\left|\left|h^2_{v_i} - h^2_{v_j}\right|\right| inf}{|E(G)|}, \text{ where } i, j \in \{1, 2, 3, 4, 5\} = 0.0448$$

3.2 Nodes Embedding

In this research we use the embedding process to reduce the dimension of river features in Probolinggo Regency. The embedding process we divide into two stages, namely the message passing stage and message aggregation stage. In the message passing process, the information contained in each node is sent to other neighboring nodes. The information collected from all neighboring nodes then goes through a summation process, this process is called message aggregation.

Through this message aggregation process, we obtain the embedding data. Then we used the embedding data to forecast which river had the most waste in the following week. In forecasting river waste data we use three stages, namely training, testing, and forecasting.

The training stage aims to train machines to be able recognize the input data. This training phase produces an output called the training model. We train the machine several times until we get the best model. The benchmark that determines whether this model is good or not is the mean square error (MSE). The smaller the MSE value, the better the model produces output values. Furthermore, the resulting model is used in the testing phase. The goal is to measure how accurate the ANN output results are. The data test results we use to predict rivers that have the potential to flood due to waste.

This study uses normalized and non-normalized data. We use iterations 8, 10, and 12, the initial weights are 0.3, 0.2, and 0.1, with $\alpha = 0.1$. The results of this experiment are shown in Table I above. In the table it can be seen that the error value in data without normalization increases with each iteration. Whereas in normalized data, the error value tends to decrease at each iteration. The comparison of the error values in Fig. 2 uses the initial weight which produces the smallest error value, namely the initial weight of 0.1.

3.3 Time Series Forecasting Analysis

Computer simulations we performed on two neural network architectures. The aim is to train, test, and forecast river waste anomaly data in Probolinggo Regency. In the previous

Table 1. The Results of GNN node embedding for finding the best loss

Data Type	Iteration Numbers	Learning Weight	Error Value
Non-Normalization	8	0.3	1.3216×10^{25}
	10	0.2	3.4377×10^{23}
	12	0.1	6.7144×10^{20}
		0.3	1.9989×10^{75}
		0.2	3.5183×10^{70}
		0.1	2.6214×10^{62}
		0.3	6.3578×10^{225}
		0.2	3.4668×10^{211}
		0.1	1.4340×10^{187}
Normalization	8	0.3	4.7137×10^{13}
	10	0.2	0.0565
	12	0.1	1.5461×10^{-4}
		0.3	2.3462×10^{31}
		0.2	0.0376
		0.1	2.2375×10^{-5}
		0.3	3.7729×10^{280}
		0.2	0.0249
		0.1	3.2262×10^{-6}

stage, time series data were obtained from the GNN embedding process. Then the data is given treatment in the form of numerical simulation using Matlab programming. There are four types of ANN models used in this study, namely Feedforwardnet, Paternnet. Fitnet, and Cascadeforwardnet. While the neural network architecture used is ANN-468 and ANN-568, using the Lavenberg-Marquadt training function and sigmoid log transfer as parameters in training the network. Figure 3 shows the neural network model and architecture that we use.

The results of the training and testing that we have done are presented in Table II. The best model indicator is the model that produces the smallest MSE (mean square error). Judging from the results of training and testing, we decide the best model is produced by ANN-468 cascadeforwardnet. In addition, plots from training, testing, and forecasting are also displayed to find out the location of anomaly data. The results of this study are shown in Fig. 4 to Fig. 7. The following week it was found that the anomaly data was in the 3rd data, which can be seen in Fig. 6. The resulting regression was fairly good, namely 0.9785, which can be seen in Fig. 7.

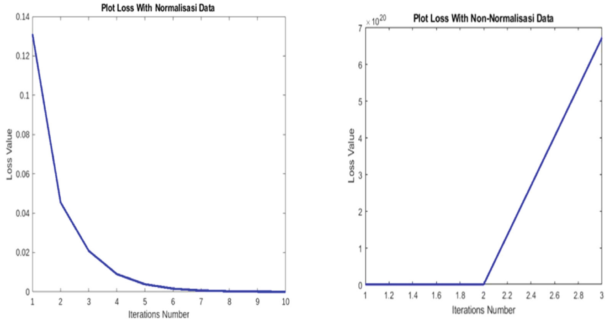

Fig. 2. Plot loss with normalized data (left) and plot loss with non-normalized data (right)

Table 2. The performance indicator of ANN architectures and models

ANN Model	ANN Architecture	MSE Train	Regression Train	Time Train	MSE TEST
Feed Forwardnet	ANN-(a) 468 ANN-(a) 568	9.3018×10^{-8} 4.4324×10^{-8}	0.11472 0.76001	2.5037 s 2.3712 s	1.0527×10^{-7} 6.0563×10^{-8}
Fitnet	ANN-(b) 468 ANN-(b) 568	4.9400×10^{-8} 2.3811×10^{-8}	0.64937 0.84152	2.465 s 2.9234 s	6.5892×10^{-8} 4.4509×10^{-8}
Paternet	ANN-(c) 468 ANN-(c) 568	4.3704×10^{-7} 3.9697×10^{-8}	0.72142 0.63041	2.3143 s 2.936 s	3.7133×10^{-7} 6.5106×10^{-8}
Cascade Forwardnet	ANN-(d) 468 ANN-(d) 568	3.6638×10^{-9} 1.4741×10^{-8}	0.9785 0.90525	4.6035 s 2.7192 s	$\mathbf{5.0687} \times 10^{-9}$ 3.5601×10^{-8}

3.4 Strong Rainbow Antimagic Coloring Application

Based on the results obtained, we have been able to predict the river pollution level in the next week. We use these results as a reference to determine the volume of automatic garbage cleaning containers. The way this tool works is by detecting approaching trash using an ultrasonic sensor. Then the signal is sent via Arduino to drive the DC motor dynamo as a garbage collector. Installation of this tool in every river of course requires a high cost. So we used the Strong Rainbow Antimagic Coloring Technique to determine the placement of this automatic river waste collection device. Figure 8 shows a graph

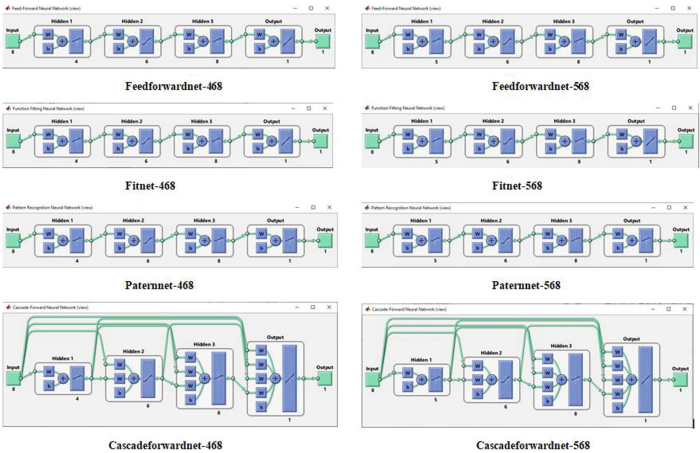

Fig. 3. The ANN model and architectures that we use

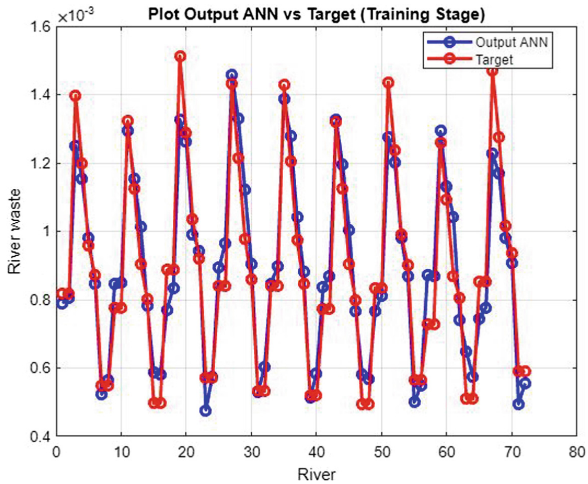

Fig. 4. The Comparison of ANN outputs and river waste target data at the training stage

of the river flow in Probolinggo which we have labeled as strong rainbow antimagic coloring.

After obtaining the labeling, we assume that river 1 ($S1$) is the main river. We construct a spanning tree graph using the strong rainbow antimagic coloring concept. That is, a spanning tree graph is formed from the edge with the shortest distance connecting $S1$ to all points on the river graph. This spanning tree graph can be seen in Fig. 9. Based on the figure, we see that there are four endpoints, namely $S2$, $S4$, $S7$, and $S8$. This end point is our reference in placing an automatic waste cleaning tool. To facilitate monitoring, we propose to use the Internet of Things. IoT is a program that connects an object with the internet. So by using a computer we can observe the amount of waste taken.

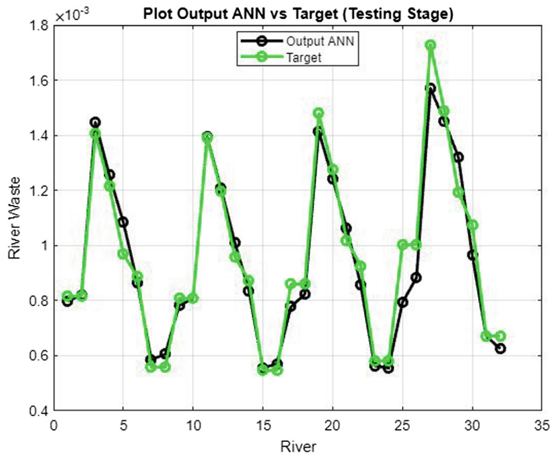

Fig. 5. The comparison of ANN output and target data of the river waste on testing stage.

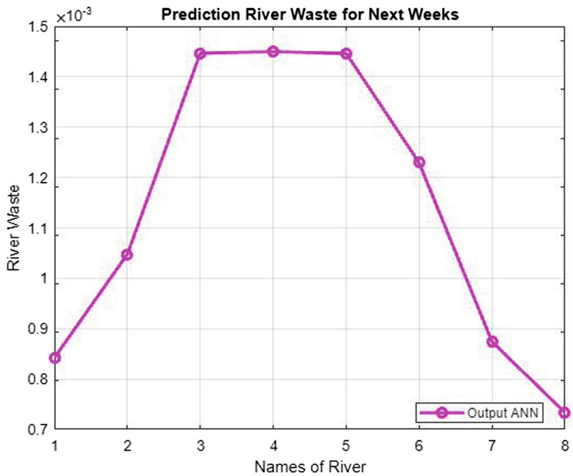

Fig. 6. Forecasting River waste to determine the river flow with the largest waste pollution.

An illustration of the placement of this automatic river waste cleaning tool can be seen in Fig. 10.

4 Result Discussion

In this research, we have succeeded in inserting a graph of waste pollution in the rivers of Probolinggo. Through the embedding process, the dimensions of the original feature can be reduced. Of course, we need to do two stages of the embedding process, namely message passing and message aggregation to get the embedding results. In the message passing process, we consider that each node has information or messages to be sent to

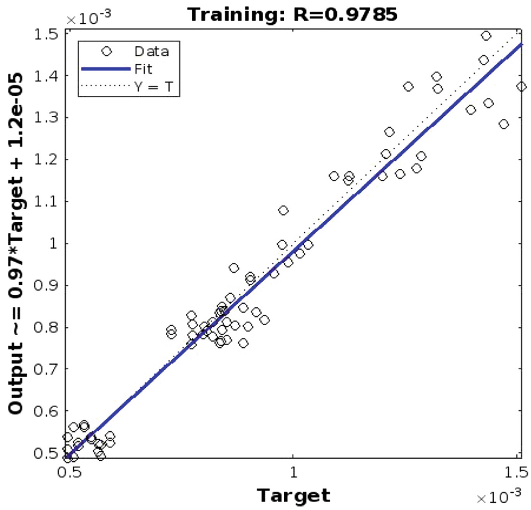

Fig. 7. The results of this regression indicate that the model we used in this study is accurate.

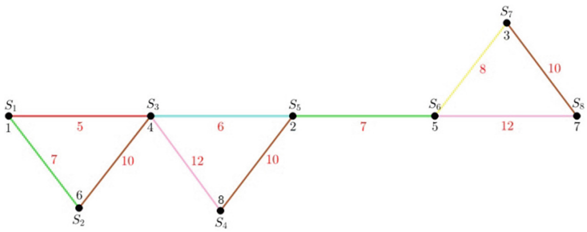

Fig. 8. The results of the river waste SRAC graph

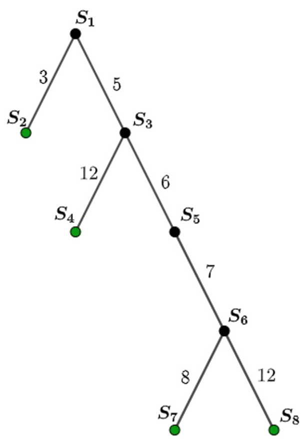

Fig. 9. Spanning Tree from a River Graph in Probolinggo

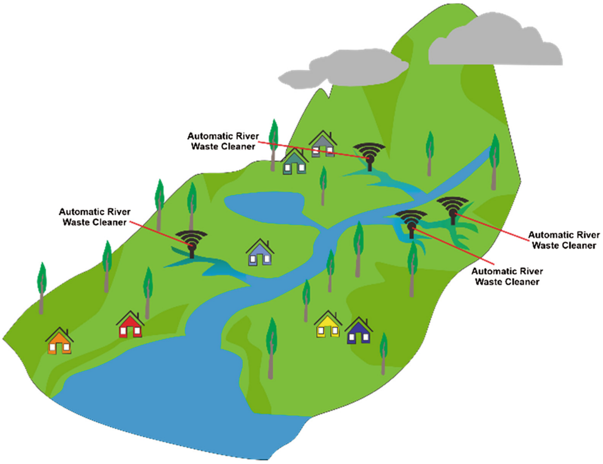

Fig. 10. The illustration placement of Automatic River Garbage Cleaner.

other neighboring nodes, so in this process we consider the adjacency matrix. However, we are not only considering the relationship of one node to adjacent nodes, but also the relationship of the node itself. Then matrix B is used as a reference in the process of sending this message. Then, we sum the messages in the aggregation process when the messages have been sent. We use the error value as a yardstick to measure how close two nodes have been processed by the aggregation. In this study, the best error value is 3.2262×10^{-6} which is generated using a weight of 0.1 and 12 iterations.

Next, we estimate the time series data using ANN after getting the embedded data. At this stage there are three parts, namely training, testing, and forecasting. There are four models and two ANN architectures that we use during the training. In network training, a model is obtained that can be used in the testing phase. The model that has been built is then tested to measure ANN performance. As a benchmark, we use the mean square error (MSE). Based on the test results, the smallest MSE 5.0687×10^{-9} obtained from ANN model Cascadeforwardnet with 468 architectures. Then by using the training model, forecasting water discharge data is obtained in the following week. This forecast shows the anomaly point is in the 3rd river which can be seen in Fig. 5.

After determining which river has the highest level of river pollution. Next we applied Strong Rainbow Antimagic Coloring (SRAC) to determine the placement of the automated river waste cleaner tools. Based on Fig. 9, we know that the number of tools needed is four. This tool is installed at the end point of the spanning tree graph in Fig. 9. An illustration of the placement of this tool can be seen in Fig. 10.

5 Conclusion

The water quality of the Pekalen River in Probolinggo Regency is polluted by household, industrial, fishery and agricultural wastes. The impacts are very dangerous, starting from river water pollution, narrowing of river bodies, high erosion and sedimentation which causes frequent flooding in the Pekalen Watershed. The efforts to overcome this problem are by cleaning the Pekalen River regular basis. Application of Spatial Temporal Graph Neural Networks (STGNN) to predict rivers with the highest waste in the next week. We are also applying the Strong Rainbow Antimagic Coloring concept to determine the installation point for the automatic river waste cleaner tools.

Acknowledgement. We would like thanks to PUI-PT Combinatorics and Graph, CGANT University of Jember, Indonesia 2023, and we are grateful for the support of Zainul Hasan Genggong Islamic University for their support and motivation in completing this paper.

References

1. D. Syurandhari, "Hubungan Perilaku Hidup Bersih dan Sehat Masyarakat dengan Kejadian Diare," *Hosp. Majapahit*, vol. 12, 2020.
2. Dafik, A'yun Q, Baihaki R I, Prihandoko A C, Kristiana A I, Febrinanto F G, and K A Santosa, 2022, "The Spatial Temporal Graph Neural Networks and Rainbow Antimagic Coloring for Time Series Forecasting on Flood Flow Anomaly" In press.
3. Dafik, F. Susanto, R. Alfarisi, B. J. Septory, I. H. Agustin, and M. Venkatachalam, "On rainbow antimagic coloring of graphs," *Adv. Math. Model. Appl.*, vol. 6, no. 3, 2021.
4. Dafik, Z. R. Ridlo, I. H. Agustin, R. I. Baihaki, F. G. Febrinanto, R. Nisviasari, Suhardi, and A. Riski, "The Implementation of Artificial Neural Networks and Resolving Efficient Dominating Set for Time Series Forecasting on Soil Moisture to Advance the Automatic Irrigation System on Vertical Farming" in press.
5. F. Ghaderi, A. H. Ghaderi, N. Ghaderi, and B. Najafi, "Prediction of the thermal conductivity of refrigerants by computational methods and artificial neural network," *Front. Chem.*, vol. 5, no. NOV 2017, doi: https://doi.org/10.3389/fchem.2017.00099.
6. J.-P. Yeh and Y.-C. Chang, "Comparison between Neural Network and Adaptive Neuro-Fuzzy Inference System for Forecasting Chaotic Traffic Volumes," *J. Intell. Learn. Syst. Appl.*, vol. 04, no. 04, 2012, doi: https://doi.org/10.4236/jilsa.2012.44025.
7. K. H. N. Bui, J. Cho, and H. Yi, "Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues," *Appl. Intell.*, vol. 52, no. 3, 2022, doi: https://doi.org/10.1007/s10489-021-02587-w
8. M. Tsodyks, A. Uziel, and H. Markram, "Synchrony generation in recurrent networks with frequency-dependent synapses.," *J. Neurosci.*, vol. 20, no. 1, 2000, doi: https://doi.org/10.1523/jneurosci.20-01-j0003.2000.
9. M. S. Hasan, Slamin, Dafik, I. H. Agustin, and R. Alfarisi, "On the total rainbow connection of the wheel related graphs," in *Journal of Physics: Conference Series*, 2018, vol. 1008, no. 1. doi: https://doi.org/10.1088/1742-6596/1008/1/012054.
10. M. Sholeh, P. Pranoto, S. Budiastuti, and S. Sutarno, "Analysis of Citarum River pollution indicator using chemical, physical, and bacteriological methods," in *AIP Conference Proceedings*, 2018, vol. 2049. doi: https://doi.org/10.1063/1.5082473.
11. P. Awasthi, A. Das, and S. Gollapudi, "A Convergence Analysis of Gradient Descent on Graph Neural Networks," in *Advances in Neural Information Processing Systems*, 2021, vol. 25.
12. R. Ekawaty, E. Gas Ekaputra, and A. Arbain, "Telaahan Daya Dukung dan Daya Tampung Lingkungan dalam Pengelolaan Kawasan Daerah Aliran Sungai di Indonesia Study of Environment Carrying Capacity in Management of Watershed in Indonesia," 2018.