BAREKENG: Journal of Mathematics and Its Applications
September 2024 Volume 18 Issue 3 Page 1615–1628
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol18iss3pp1615-1628

ANALYSIS OF RESOLVING EFFICIENT DOMINATING SET AND ITS APPLICATION SCHEME IN SOLVING ETLE PROBLEMS

RM Prihandini [1*], MR Rahmadani [2], Dafik[3]

[1,2,3]*Mathematics Education Program, Faculty of Teacher Training and Education, Universitas Jember Kampus Tegal Boto, Jember, 68121, Indonesia*

*Corresponding author's e-mail: *rafiantikap.fkip@unej.ac.id*

ABSTRACT

This research focuses on the analysis of Resolving Efficient Dominating Set (REDS) and its application in solving Electronic Traffic Law Enforcement (ETLE) problems using the Spatial Temporal Graph Neural Network (STGNN). Resolving Efficient Dominating Set (REDS) is a concept in graph theory that studies a set of points in a graph that efficiently monitors other points. It involves ensuring that each point $v \in V(G) - D$ is dominated by exactly one point in D, with no adjacent points in D, and the representation of point $v \in V(G)$ concerning D is not the same, which is termed as a resolving efficient dominating set. In the context of Electronic Traffic Law Enforcement (ETLE), the analysis of REDS has a significant impact. The theorem resulting from the analysis of REDS enables the determination of the number of traffic violation sensors required. Furthermore, by taking simulation data from road points, violation forecasting can be performed. The accurate predictions from this forecasting can assist authorities in anticipating and addressing traffic violation issues more effectively.

Article History:

Received: 1st January 2024
Revised: 1st April 2024
Accepted: 27th June 2024
Published: 1st September 2024

Keywords:

Resolving efficient dominating set;
Electronic traffic law enforcement;
Graph Neural Network.

How to cite this article:

R. M. Prihandini, M. R. Rahmadani, and Dafik., "ANALYSIS OF RESOLVING EFFICIENT DOMINATING SET AND ITS APPLICATION SCHEME IN SOLVING ETLE PROBLEMS," *BAREKENG: J. Math. & App.*, vol. 18, iss. 3, pp. 1615-1628, September, 2024.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

Traffic is the movement of vehicles on the road from one place to another. Traffic violations are the most common occurrences on the roads, committed by a majority of the population, including users of four-wheeled vehicles, two-wheeled vehicles, buses, or trucks. This is the primary concern on the roads and is a crucial task for the police [1], [2]. In the implementation of ETLE (Electronic Traffic Law Enforcement), the police deploy traffic violation detection sensors at each road point. The Electronic Traffic Law Enforcement system requires a vital management center to enhance the monitoring and enforcement of traffic rules efficiently, namely the traffic control center. This traffic control center is a location that enables the monitoring of data received from road network sensors and manages the handling of traffic violations. In placing the traffic control center for Electronic Traffic Law Enforcement, the location and operational costs will be strategic if kept to a minimum while still being able to effectively monitor and enforce traffic laws in various specific zones. Therefore, a method is needed to place the traffic control center, one of which is by applying a branch of mathematics, namely graph theory.

A graph G is an ordered pair of sets $V(G)$ and $E(G)$, where $V(G)$ is a non-empty set of elements called vertices, and $E(G)$ is a (possibly empty) set of unordered pairs of elements from $V(G)$, called edges [3]. From this definition, a graph can be formed with at least one vertex and may not necessarily have edges. A graph with at least one vertex and no edges is called an empty graph. The number of vertices in a graph is referred to as the order of G. The number of edges in the graph G is called the size of G. The order of the graph G is denoted by $|V(G)|$, and the size of the graph G is denoted by $|E(G)|$ [4].

In this research, the chosen graph theory topic is *Resolving Efficient Dominating Set*. A resolving efficient dominating set exists if every point $v \in V(G) - D$ is dominated by exactly one point in D, and no two points in D are adjacent, and the representation of point $v \in V(G)$ to D is unique [5], [6]. Some relevant research can be seen at [7], [8], [9], [10], [11], [12], [13], [14]. These studies provide valuable insights into the scientific analysis of Resolving Efficient Dominating Set (REDS), exploring the minimum cardinality required to efficiently dominate points within a graph, thus contributing to the advancement of graph theory. $\gamma_{re}(G)$ represents the *Resolving Efficient Dominating Number* of graph G, which is the minimum cardinality required to efficiently monitor other points. Resolving efficient dominating set in this study will be linked to the analysis of Electronic Traffic Law Enforcement (ETLE). The objective of this analysis is to utilize the resolving efficient dominating set method in Electronic Traffic Law Enforcement (ETLE) to optimize the deployment of monitoring devices, thereby improving traffic management and law enforcement efficiency. To solve the ETLE problem in traffic using the concept of resolving efficient dominating set, the road network map will be represented as a graph. By using the resolving efficient dominating set algorithm, the ETLE system can determine an efficient dominating set that minimizes the number of monitoring devices required.

The resolving efficient dominating set is applied to identify the most efficient and strategic locations where traffic control centers are placed to monitor data and manage the handling of traffic violations. By using the resolving efficient dominating set algorithm, the ETLE system can determine an efficient dominating set that minimizes the number of such control centers. Using the resolving efficient dominating set algorithm, the ETLE system can determine an efficient dominating set that minimizes the number of traffic control center devices needed. This placement will be based on the concept of a distinguishing efficient dominating set, where each point in a graph is dominated by exactly one member of the distinguishing efficient dominating set. A Graph Neural Network is utilized to classify each node in a graph and discern the meaning of each node [15]. STGNN is a neural network capable of processing spatial and temporal data, aiding in analyzing how systems or phenomena change over time represented in graph form. In this paper, a Spatial Temporal Graph Neural Network (STGNN) assisted with Python is used in solving Electronic Traffic Law Enforcement (ETLE) to determine the number of traffic violations that occur within a specific time frame.

2. RESEARCH METHODS

This research employs analytical and experimental methods. In the analytical study axiomatic deductive method, which employs pre-existing deductive principles in mathematical logic with existing axioms or theorems, is then applied to a problem related to resolving an efficient dominating set. In the experimental method, we use STGNN (Spatial Temporal Graph Neural Network) which is a method that uses a special type of computer program to understand how things change over both space and time, helping us to better understand and predict patterns in data that evolve over both dimensions.

In this paper, we will analyze traffic violations in Ponorogo City, East Java. There are five features utilized in the vertex embedding process, including the number of riders not wearing helmets, violating road markings, going against traffic flow, running red lights, and not turning on headlights. Subsequently, in the STGNN programming, the model is trained using the obtained data, tested, and ultimately used to predict traffic violations in Ponorogo City.

We employ the following algorithm to study traffic violations using STGNN combined with a resolving efficient dominating set:

1. Given a graph $G(V, E)$ of order n and a feature matrix $H_{n\times m}$ of n point and m features, and given tolerance ϵ;
2. Determining the adjacency matrix A of the graph G and defining the matrix $B = A + I$, where I is the identity matrix;
3. Initialize the weight Wg, bias β, and learning rate α. (For simplicity, set $Wg_{m\times 1} = [w_1 w_2 \dots w_m]$, where $0 \leq w_j \leq 1$, bias $\beta = 0$, and $0 \leq \alpha \leq 1$);
4. Multiply the weight matrix by the point features, by defining the *message* function $m_u^l = MSG^l\left(h_u^{i-1}\right)$. For the linear layer $m_u^l = Wg^l\left(h_u^{i-1}\right)$. In the equation, l refers to the layer index, u denotes the node being processed, and i represents the iteration or step;
5. Sum the messages from neighboring points, using the function $h_x^l = AGG^l m_u^{l-1}, u \in N(v)$ and apply the **sum(.)** function, $h_x^l = sum^l m_u^{l-1}, u \in N(v)$;
6. Determine the *error*, where $error^l = \frac{\left\|h_{xi}^l - h_{xj}^l\right\|}{|E(G)|^2}$, where v_i, v_j are two adjacent points;
7. Observe whether the $error \leq \epsilon$. yes, it means to stop; if not, proceed to step 8 to update the weight matrix *W*;
8. Update the weight matrix with $Wg^{l+1} = W_j^l + \alpha \times z_j \times e^l$ where z_j is the sum of each column in H_{xi}^l divided by the number of *nodes*;
9. Repeat steps 4-7 until $error \leq \epsilon$;
10. Obtain *training*, *testing*, and *forecasting* results, then stop.

3. RESULTS AND DISCUSSION

3.1 Research Findings on Resolving Efficient Dominating Set

This research produces four theorems regarding the resolving efficient domination number. Below are the theorem results along with the proofs regarding the resolving efficient domination number in the graph $P_n \odot P_2$, $S_n \odot P_2$, $C_n \odot P_2$, and $Shack(G, u_i, n)$. The corona $G_1 \odot G_2$ of two graphs G_1 (with n_1 vertices and m_1 edges) and G_2 (with n_2 vertices and m_2 edges) is defined as the graph obtained by taking one copy of G_1 and n_1 copies of G_2, and then joining the *i*-th vertex of G_1 with an edge to every vertex in the *i*-th copy of G_2. The notation $\gamma_{re}(G)$ is used to determine the minimum number required to efficiently dominate the graph G.

Theorem 1. For the graph $P_n \odot P_2$ with $n \geq 2$, $\gamma_{re}(P_n \odot P_2) = n$.

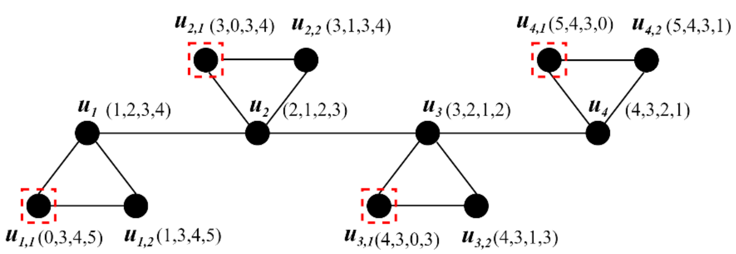


Figure 1. Resolving Efficient Dominating Set ($P_4 \odot P_2$)

Proof. The graph $P_n \odot P_2$ is a graph with the vertex set $V(P_n \odot P_2) = \{u_i; 1 \le i \le n\} \cup \{u_{i,j}; 1 \le i \le n, 1 \le j \le 2\}$ and the edge set $E(P_n \odot P_2\,) = \{u_i\, u_{i+1}; 1 \le i \le n-1\} \cup \{u_i\, u_{i,j}; 1 \le i \le n, 1 \le j \le 2\} \cup \{u_{i,1}\, u_{i,2}; 1 \le i \le n\}$. The cardinality of the vertex set and edge set of the graph $P_n \odot P_2$ *is* $|V(P_n \odot P_2)| = 3n$ and $|E(P_n \odot P_2\,)| = 4n - 1$.

Based on **Figure 1** it depicts the result of the corona operation of a path graph of order 4 (P_4) with a path graph of order 2 (P_2). In the P_4 graph, the P_2 graph is duplicated 4 times, and then the points in each P_2 graph is connected to every point in the P_4 graph.

We will prove that $\gamma_{re}\,(P_n \odot P_2) = n$ by demonstrating the upper bound $\gamma_{re}\,(P_n \odot P_2)$ and the lower bound $\gamma_{re}\,(P_n \odot P_2)$. First, let's establish the upper bound for the resolving efficient domination number of $P_n \odot P_2$. Assuming $|D_1| = n + 1$, we obtain:

(i) If $D_1 = \{u_{i,1}; 1 \le i \le n\} \cup \{u_i\}$, there will be a point $u_{i,2}$ where $|N(u_{i,2}) \cap D_1| \ne 1$. Thus, D_1 is not an efficient dominating set;

(ii) If $D_1 = \{u_{i,2}; 1 \le i \le n\} \cup \{u_i\}$, there will be a point $u_{i,1}$ where $|N(u_{i,1}) \cap D_1| \ne 1$. Thus, D_1 is not an efficient dominating set.

Secondly, we choose $D_2 = \{u_{i,1};\ 1 \le i \le n\}$ in such a way that $|D_2| = n$. Then, to determine if the representation of each vertex with respect to D_2 is distinct from each other, we can examine the distance function between two points in the graph $P_n \odot P_2$, which is:

$$d(u_i u_{k,1}) = |i-k| + 1$$

$$d(u_{i,1} u_{k,1}) = \begin{cases} |i-k| + 2, & i \ne k \\ 0 & i = k \end{cases}$$

$$d(u_{i,1} u_{k,1}) = \begin{cases} |i-k| + 2, & i \ne k \\ 1 & i = k \end{cases}$$

D_2 is an efficient dominating set because every point $u_i, u_{i,j} \in V(P_n \odot P_2\,) - D_2$ is dominated by exactly one point $u_{i,1} \in D_2$, and $N(u_i) \cap D_2 = u_{i,1}, N(u_{i,2}) \cap D_2 = u_{i,1}$, ensuring that $|N(u) \cap D_2| = 1$. Therefore, D_2 is a resolving efficient dominating set.

Third, we will prove the lower bound for $P_n \odot P_2$. Assuming $|D_3| = n - 1$, we obtain:

(i) If $D_3 = \{u_{i,1};\ 1 \le i \le n-1\}$, there are points u_i and u_{ij} that are not dominated by $u_{i,1}$ in D_3. Thus, D_3 is not an efficient dominating set;

(ii) If $D_3 = \{u_{i,2};\ 1 \le i \le n-1\}$, there are points u_i and u_{ij} that are not dominated by $u_{i,2}$ in D_3. Thus, D_3 is not an efficient dominating set;

(iii) If $D_3 = \{u_i;\ 1 \le i \le n-1\}$, there are points u_i where $|N(u_i) \cap D_3\,| \ne 1$. Thus, D_3 is not an efficient dominating set.

Based on these proofs, we can conclude that $|D_2|$ is the minimum cardinality of the resolving efficient dominating set in the graph $P_n \odot P_2$, hence $\gamma_{re}\,(P_n \odot P_2\,) = n$. ∎

Theorem 2. For the star graph S_n with $n \ge 2$, and the path graph P_2, $\gamma_{re}\,(S_n \odot P_2\,) = n + 1$.

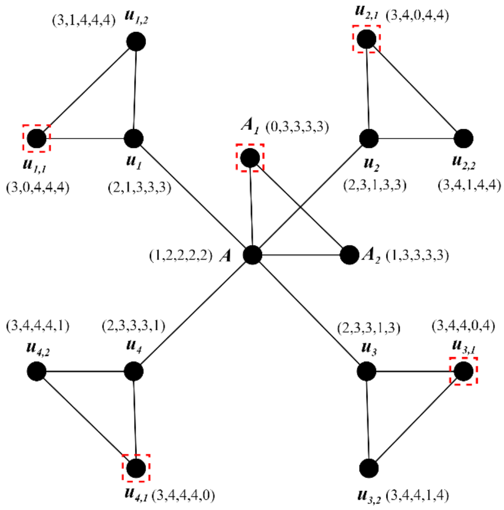


Figure 2. **Resolving Efficient Dominating Set ($S_4 \odot P_2$)**

Proof. The set of vertices and the set of edges in the graph $S_n \odot P_2$ are $V(S_n \odot P_2) = \{A\} \cup \{u_i; 1 \le i \le n\} \cup \{A_i; 1 \le i \le 2\} \cup \{u_{i,j}; 1 \le i \le n, 1 \le j \le n\}$, $E(S_n \odot P_2) = \{Au_i; 1 \le j \le n\} \cup \{AA_i; 1 \le j \le 2\} \cup \{A_1A_2\} \cup \{u_iu_{i,j}; 1 \le i \le n, 1 \le j \le 2\} \cup \{u_{i,1}u_{i,2}; 1 \le i \le n\}$. The cardinality of the set of vertices and the set of edges of the graph $S_n \odot P_2$ is $|V(S_n \odot P_2)| = 3n + 3$ and $|E(S_n \odot P_2)| = 4n + 3$.

$\gamma_{re}\ (S_n \odot P_2) = n + 1$ is proven by showing the upper bound $\gamma_{re}\ (S_n \odot P_2\ ) \le n + 1$ and the lower bound $\gamma_{re}\ (C_n \odot P_2) \ge n + 1$.

First, we prove the upper bound of the resolving efficient domination number of $S_n \odot P_2$. Assuming $|D| = n + 2$, we obtain:

(i) If $D_1 = \{u_{i,1}; 1 \le i \le n\} \cup \{A_1\} \cup \{A\}$, then there will be a point u_i such that $|N(u_i) \cap D_1| \ne 1$ and a point A_2 such that $|N(A_2) \cap D_1| \ne 1$ Thus, D_1 is not an efficient dominating set;

(ii) If $D_1 = \{u_{i,1}; 1 \le i \le n\} \cup \{A\} \cup \{u_i\}$, there will be a point $u_{i,1}$ where $|N(u_{i,1}) \cap D_1| \ne 1$. Thus, D_1 is not an efficient dominating set.

Second, we choose $D_2 = \{u_{i1}; 1 \le i \le n\} \cup \{A_1\}$ such that $|D_2| = n + 1$. To determine whether the representation of each node with respect to D_2 is distinct from one another, we can refer to Table 1. D_2 is an efficient dominating set because each point $A, A_i, u_i, u_{ij} \in V(P_n \odot P_2) - D_2$ is dominated by exactly one point $u_{i,1}, A_1 \in D_2$, and $N(u_i) \cap D_2 = u_{i,1}, N(u_{i,2}\ ) \cap D_2 = u_{i,1}, N(A) \cap D_2 = A_1,\ N(A_2) \cap D_2 = A_1$, so $|N(u) \cap D_2 = 1|$ and $|N(A) \cap D_2 = 1|$. Thus, D_2 is a resolving efficient dominating set.

Table 1. **Representation Of Points u And A in $V(S_n \odot P_2)$ With Respect To D_2**

n	$r(v\|D)$	Condition
u_1	$(2,1,\underbrace{3,\dots,3}_{n-1})$	$n \ge 2$
u_i	$(2,\underbrace{3,\dots,3}_{i-1},1,\underbrace{3,\dots,3}_{n-1})$	$i \ge 2$
u_n	$(2,\underbrace{3,\dots,3}_{n-1})$	$n \ge 2$
$u_{1,1}$	$(3,0,\underbrace{4,\dots,4}_{n+2})$	$n \ge 2$
$u_{i,j}$	$(3,\underbrace{4,\dots,4}_{i-1},0,\underbrace{4,\dots,4}_{n-i})$	$i \ge 2, j = 1$
$u_{n,j}$	$(3,\underbrace{4,\dots,4}_{n-1},0)$	$n \ge 2, j = 1$
$u_{1,2}$	$(3,1,\underbrace{4,..,4}_{n-1})$	$n \ge 2$
$u_{i,j}$	$(3,\underbrace{4,\dots,4}_{i-1},1,\underbrace{4,\dots,4}_{n-i})$	$i \ge 2, j = 2$

n	$r(v\|D)$	Condition
$u_{n,j}$	$(3,\underbrace{4,..,4}_{n-1},1)$	$n \geq 2, j = 2$
A	$(1,\underbrace{2,..,2}_{n})$	$n \geq 2$
A_1	$(0,\underbrace{3,..,3}_{n})$	$n \geq 2$
A_2	$(1,\underbrace{3,..,3}_{n})$	$n \geq 2$

Third, we will prove the lower bound of $\gamma_{re}\ (S_n \odot P_2)$. Assuming $|D_3| = n$, we obtain:

(i) If $D_3 = \{u_{i,1}; 1 \leq i \leq n\}$, then there exist points A and A_i that are not dominated by $u_{i,1}$ in D_3. Thus, D_3 is not an efficient dominating set;

(ii) If $D_3 = \{u_{i,2}; 1 \leq i \leq n\}$, then there exist points A and A_i that are not dominated by $u_{i,2}$ in D_3. Thus, D_3 is not an efficient dominating set.

(iii) If $D_3 = \{u_i; 1 \leq i \leq n\}$, then there exists a point u_i where $|N(u_i) \cap D_3\,| \neq 1$. Thus, D_3 is not an efficient dominating set;

(iv) *If* $D_3 = \{u_i; 1 \leq i \leq n-1\} \cup \{A\}$, then there exists a point u_i where $|N(u_i) \cap D_3\,| \neq 1$, and there exist points A_1 and $A_2 \in V(S_n \odot P_2)$ such that $r(A_1 \,|\, D_3) = r(A_1 \,|\, D_3) = (1,2, \dots, 2)$. Thus, D_3 is not an efficient dominating set

Based on these proofs, we can conclude that $|D_2|$ is the minimum cardinality of a resolving efficient dominating set in the graph $S_n \odot P_2$ such that $\gamma_{re}\ (S_n \odot P_2) = n + 1$. ■

Theorem 3. For the cycle graph C_n *and the path graph* P_2, $\gamma_{re}\ (C_n \odot P_2) = n$, *for* $n \geq 3$.

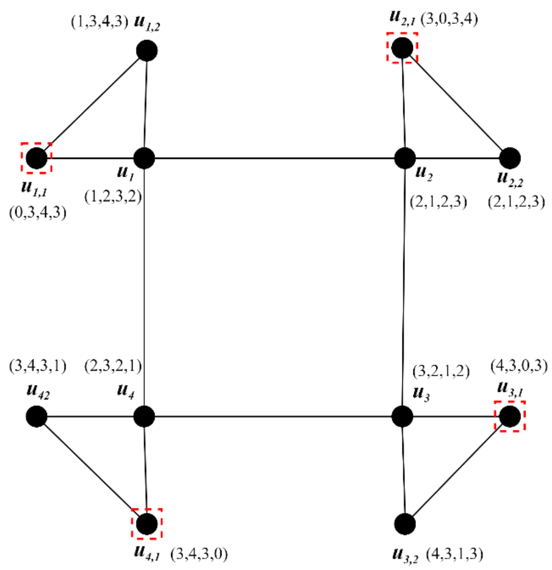


Figure 3. Resolving Efficient Dominating Set ($C_4 \odot P_2$)

Proof. The graph $C_n \odot P_2$ is the corona graph between the cycle graph C_n with $n \geq 3$ and the path graph of order 2 (P_2). The set of vertices of $C_n \odot P_2$ is denoted as $V(C_n \odot P_2) = \{u_i; 1 \leq i \leq n\} \cup \{u_{i,j}; 1 \leq i \leq n, 1 \leq j \leq 2\}$, and the set of edges is denoted as $E(C_n \odot P_2) = \{u_i\, u_{i+1}; 1 \leq i \leq n-1\} \cup \{u_1\, u_n\,\} \cup \{u_i\, u_{i,j}; 1 \leq i \leq n, 1 \leq j \leq 2\} \cup \{u_{i,1}\, u_{i,2}; 1 \leq i \leq n\}$. The cardinality of the set of vertices and the set of edges of the graph $C_n \odot P_2$ are $|V(C_n \odot P_2)| = 3n$ and $|E(C_n \odot P_2)| = 4n$.

The proof of $\gamma_{re}\ (C_n \odot P_2\,) = n$ is demonstrated by establishing the upper bound $\gamma_{re}\ (C_n \odot P_2) \leq n$ and the lower bound $\gamma_{re}\ (C_n \odot P_2\,) \geq n$. First, we prove the resolving efficient domination number bounds of $C_n \odot P_2$. Assuming $|D_1| = n + 1$, we obtain:

(i) If $D_1 = \{u_{i,2}; 1 \leq i \leq n\} \cup \{u_i\}$, there will be a point $u_{i,1}$ where $|N(u_{i,1}) \cap D_1| \neq 1$. Thus, D_1 is not an efficient dominating set;

(ii) If $D_1 = \{u_{i,1}; 1 \leq i \leq n\} \cup \{u_i\}$, there will be a point $u_{i,2}$ where $|N(u_{i,2}) \cap D_1| \neq 1$. Thus, D_1 is not an efficient dominating set.

Secondly, we choose $D_2 = \{u_{i,1}; 1 \leq i \leq n\}$ such that $|D_2| = n$. Each point $u_i, u_{i,j} \in V(C_n \odot P_2) - D_2$ is dominated by exactly one point $u_{i,1} \in D_2$, and $N(u_i) \cap D_2 = u_{i,1}$, $N(u_{i,2}) \cap D_2 = u_{i,1}$, so $|N(u) \cap D_2| = 1$, thereby making D_2 an efficient dominating set. Then, to determine whether the representation of each vertex with respect to D_2 is distinct from one another, we can examine the distance function between two points in the graph $C_n \odot P_2$, namely:

$$d(u_i u_{k,1}) = |i-k| + 1$$

$$d(u_{i,1} u_{k,1}) = \begin{cases} |i-k| + 3, & i \neq k \\ 0 & i = k \end{cases}$$

$$d(u_{i,1} u_{k,1}) = \begin{cases} |i-k| + 3, & i \neq k \\ 1 & i = k \end{cases}$$

So, D_2 is a resolving efficient dominating set.

Third, we will prove the lower bound of γ_{re} $(C_n \odot P_2)$. Assuming $|D_3| = n - 1$, we obtain:

(i) If $D_3 = \{u_{i,1};\ 1 \leq i \leq n-1\}$, there will be points u_i and u_{ij} that are not dominated by $u_{i,1}$ in D_3. Thus, D_3 is not an efficient dominating set;

(ii) If $D_3 = \{u_{i,2};\ 1 \leq i \leq n-1\}$, there will be points u_i and u_{ij} that are not dominated by $u_{i,2}$ in D_3. Thus, D_3 is not an efficient dominating set;

(iii) If $D_3 = \{u_i;\ 1 \leq i \leq n-1\}$, there will be a point u_i where $|N(u_i) \cap D_3| \neq 1$. Thus, D_3 is not an efficient dominating set.

Based on these proofs, we can conclude that $|D_2|$ is the minimum cardinality of the resolving efficient dominating set in the graph $C_n \odot P_2$, hence γ_{re} $(C_n \odot P_2) = n$. ■

Theorem 4. *If the $Shack(G, u_i, n)$ graph is a shackle operation graph with $n \geq 2$, then* γ_{re} $(Shack(G, u_i, n)) = 2n + 1$.

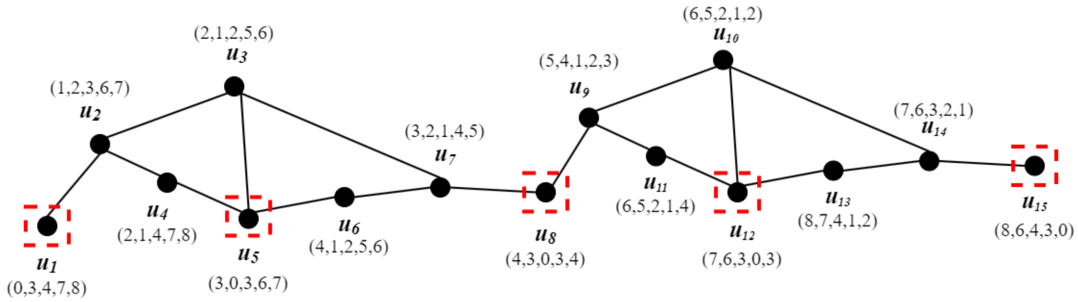


Figure 4. Resolving Efficient Dominating Set ($Shack(G, u_i, 2)$)

Proof. G represents the graph representation of the road map of Ponorogo, with u_i representing the points corresponding to intersections or critical points in Ponorogo. Subsequently, we can associate the shackle operation on this representation graph with duplication or replication n times. The $Shack(G, u_i, n)$ graph is the shackle operation graph of the representation graph of the Ponorogo road map with order 8. The $Shack(G, u_i, n)$ graph has $7n + 1$ vertices, and $|D| = 2n + 1$.

It will be proven that γ_{re} $(Shack(G, u_i, n)) = 2n + 1$ by demonstrating the upper bound of γ_{re} $(Shack(G, u_i, n))$ and the lower bound of γ_{re} $(Shack(G, u_i, n))$. First, we prove the upper bound of the resolving efficient domination number of $Shack(G, u_i, n)$. Assuming $|D_1| = 2n + 2$ we obtain if $D_1 = \{u_i; 1 \leq i \leq 2n + 2\}$ then there exists a point u_i where $|N(u_i) \cap D_1| \neq 1$. Thus D_1 is not an efficient dominating set.

If $D_1 = \{u_i; 1 \leq i \leq 2n + 2\}$ then there exists a point u_i where $|N(u_i) \cap D_1| \neq 1$. Thus D_1 is not an efficient dominating set;

Second, assume $|D_2| = 2n + 1$. D_2 is an efficient dominating set because each point $u_i \in V(Shack(G, u_i, n)) - D_2$ is dominated by exactly one point $u_i \in D_2$. Furthermore, we will show that the chosen D_2 also satisfies the characteristics of a resolving set. To determine whether the representation of each node with respect to D_2 is distinct from one another, we can examine the distance function between two nodes in the graph $Shack(G, u_i, n)$:

$$d(u_i u_k) = \begin{cases} |i-k|, & i = k \\ |i-k|+1, & i \neq k \end{cases}$$

So, D_2 is a resolving efficient dominating set.

Third, we will prove the lower bound of $Shack(G, u_i, n)$. Assume $|D_3| = n + 1$ we obtain if $D_3 = \{u_i; 1 \leq i \leq n+1\}$, then there exists a point u_i that is not dominated by any u_i in D_1. Thus, D_3 is not an efficient dominating set.

If $D_3 = \{u_i; 1 \leq i \leq n+1\}$, then there exists a point u_i that is not dominated by any u_i in D_1. Thus, D_3 is not an efficient dominating set;

Based on these proofs, we can conclude that $|D_2|$ is the minimum cardinality of a resolving efficient dominating set in the graph $Shack(G, u_i, n)$, hence $\gamma_{re}(Shack(G, u_i, n)) = 2n + 1$. ∎

3.2. The Implementation of The Resolving Efficient Dominating Set in Solving Electronic Traffic Law Enforcement Problems Using STGNN

The analysis of *resolving an efficient dominating set* in the electronic traffic law enforcement problem aims to enhance the efficiency of electronic traffic law enforcement. A road network is represented as a graph. Subsequently, determining the dominator of the road map graph with the condition that each point in the graph is dominated by only one dominator and is not adjacent to each other. Then, determining the representation of each point with respect to the dominator, if the representation of each point is different, it meets the criteria of a resolving efficient dominating set. If there are two points with the same distance representation, the activity is repeated by searching for the dominator's location again. Resolving an efficient dominating set will be interpreted as an operator monitoring traffic violations in specific zones. The concept of resolving an efficient dominating set is applied to identify the most efficient and strategic locations where traffic surveillance operators or devices should be placed to monitor and control traffic efficiently. In other words, dominant nodes (dominators) in the road network graph are interpreted as optimal locations for traffic violation surveillance operators.

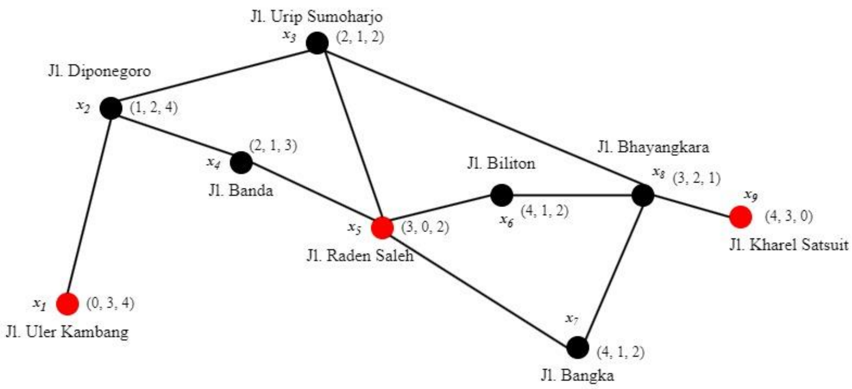


Figure 5. REDS Represents the Graph Representation of the Road Network in Ponorogo

The road map of Ponorogo depicts a network of roads traversing the city from north to south and from east to west, seamlessly connecting every corner of the city. Each road in the map becomes a node in the graph. The edges between these nodes represent the connections or roads linking two points. For instance, a highway between two intersections would be an edge in the graph. **Figure 5** represents the basic graph in **Theorem 4**. Based on **Figure 5** the result of the resolving efficient domination number is three. From this result, there are three optimal locations for ETLE traffic control centers to conduct surveillance effectively and respond to violations. This research utilizes simulated traffic violation data. The traffic violation data used consists of 5 features: driving without a helmet, violating lane markings, driving against traffic, running red lights, and driving without headlights. The study focuses on 8 road points in Ponorogo City and spans a duration of 60 days. The obtained data is initially non-normalized and is subsequently normalized.

The resolution of the electronic traffic law enforcement (ETLE) problem is carried out using graph neural network (GNN) techniques. Numerical calculations are performed first to demonstrate the execution of the graph neural network (GNN) algorithm. These numerical calculations are conducted when dealing with a small-sized dataset. Given a graph G with order 8, the set of vertices and edges of graph G are $V\ (G) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$ and $E(G) = \{x_1x_2, x_2x_3, x_2x_4, x_3x_5, x_4x_5, x_3x_7, x_5x_6, x_6x_7, x_7x_8\}$. Feature matrix $H_{n\times m}$ of 8 point and 5 features are given as follows:

$$h_{vi} = \begin{bmatrix} 0.446666667 & 0.333333333 & 0.333333333 & 0.1 & 0.291304348 \\ 0.1 & 0.9 & 0.9 & 0.569565217 & 0.465217391 \\ 0.9 & 0.6 & 0.6 & 0.9 & 0.62173913 \\ 0.54 & 0.45 & 0.45 & 0.447826087 & 0.9 \\ 0.633333333 & 0.766666667 & 0.766666667 & 0.760869565 & 0.795652174 \\ 0.9 & 0.55 & 0.55 & 0.552173913 & 0.569565217 \\ 0.62 & 0.1 & 0.1 & 0.308695652 & 0.1 \\ 0.366666667 & 0.233333333 & 0.233333333 & 0.239130435 & 0.204347826 \end{bmatrix}$$

With the obtained graph, we can determine the adjacency matrix, identity matrix, and loop-adjacency matrix as follows:

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

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

We initiate the technical calculations by initializing the learning weights from a matrix (5, 5), which means the number of hidden layers is five, and the number of neurons in each hidden layer is five.

$$W = \begin{bmatrix} 0.02 & 0.05 & 0.03 & 0.1 & 0.04 \\ 0.01 & 0.06 & 0.08 & 0.02 & 0.05 \\ 0.03 & 0.07 & 0.02 & 0.09 & 0.06 \\ 0.04 & 0.01 & 0.07 & 0.05 & 0.02 \\ 0.08 & 0.05 & 0.04 & 0.03 & 0.06 \end{bmatrix}$$

And $\epsilon = 0{,}001$. The first iteration $m_{vi}^{l+1} = h_{vi}.\ W^l,\ l = 0,1,2,3,4,\cdots,n$ can be explained as follows:

$$m_{vi}^1 = h_{vi}^0.\ W^0$$

$$m^1{}_{vi} = \begin{bmatrix} 0.04957101 & 0.08123188 & 0.06538551 & 0.09507246 & 0.07401159 \\ 0.098 & 0.15095652 & 0.15147826 & 0.15143478 & 0.14230435 \\ 0.12773913 & 0.16308696 & 0.17486957 & 0.21965217 & 0.15730435 \\ 0.11871304 & 0.13497826 & 0.12854783 & 0.1528913 & 0.13405652 \\ 0.13742029 & 0.17872464 & 0.18075362 & 0.20957971 & 0.17262319 \\ 0.10765217 & 0.1505 & 0.14343478 & 0.19519565 & 0.14171739 \\ 0.03674783 & 0.05208696 & 0.0542087 & 0.09143478 & 0.04797391 \\ 0.04257971 & 0.06127536 & 0.05924638 & 0.08042029 & 0.05737681 \end{bmatrix}$$

Considering matrix B and applying the **sum(·)** aggregation on the elements of matrix m_{vi}^1, we have:

$$h_{v1,1}^1 = \sum[0.04957101;\ 0.098] = 0.14757101$$
$$h_{v2,1}^1 = \sum[0.04957101;\ 0.098;\ 0.12773913;\ 0.11871304] = 0.39402318$$
$$h_{v3,1}^1 = \sum[0.098;\ 0.12773913;\ 0.13742029;\ 0.03674783] = 0.40090725$$
$$h_{v4,1}^1 = \sum[0.098;\ 0.11871304;\ 0.13742029] = 0.35413333$$
$$h_{v5,1}^1 = \sum[0.12773913;\ 0.11871304;\ 0.13742029;\ 0.10765217] = 0.49152463$$
$$h_{v6,1}^1 = \sum[0.13742029;\ 0.10765217;\ 0.03674783] = 0.28182029$$
$$h_{v7,1}^1 = \sum[0.12773913;\ 0.10765217;\ 0.03674783;\backslash 0.04257971] = 0.31471984$$
$$h_{v8,1}^1 = \sum[0.03674783;\ 0.04257971] = 0.07932754$$
$$h_{v1,2}^1 = \sum[0.08123188;\ 0.15095652] = 0.2321884$$
$$h_{v2,2}^1 = \sum[0.08123188;\ 0.15095652;\ 0.16308696;\ 0.13497826] = 0.53025362$$
$$h_{v3,2}^1 = \sum[0.15095652;\ 0.16308696;\ 0.17872464;\ 0.05208696] = 0.54485508$$
$$h_{v4,2}^1 = \sum[0.15095652;\ 0.13497826;\ 0.17872464] = 0.46465942$$
$$h_{v5,2}^1 = \sum[0.16308696;\ 0.13497826;\ 0.17872464;\ 0.1505] = 0.62729$$
$$h_{v6,2}^1 = \sum[0.17872464;\ 0.1505;\ 0.05208696] = 0.3813116$$
$$h_{v7,2}^1 = \sum[0.16308696;\ 0.1505;\ 0.05208696;\ 0.06127536] = 0.42694928$$
$$h_{v8,2}^1 = \sum[0.05208696;\ 0.06127536] = 0.11336232$$
$$h_{v1,3}^1 = \sum[0.06538551;\ 0.15147826] = 0.21686377$$
$$h_{v2,3}^1 = \sum[0.06538551;\ 0.15147826;\ 0.17486957;\ 0.12854783] = 0.52028117$$
$$h_{v3,3}^1 = \sum[0.15147826;\ 0.17486957;\ 0.18075362;\ 0.0542087] = 0.56131015$$
$$h_{v4,3}^1 = \sum[0.15147826;\ 0.12854783;\ 0.18075362] = 0.46077971$$
$$h_{v5,3}^1 = \sum[0.17486957;\ 0.12854783;\ 0.18075362;\ 0.14343478] = 0.6276058$$
$$h_{v6,3}^1 = \sum[0.18075362;\ 0.14343478;\ 0.0542087] = 0.3783971$$
$$h_{v7,3}^1 = \sum[0.17486957;\ 0.14343478;\ 0.0542087;\ 0.05924638] = 0.43175943$$
$$h_{v8,3}^1 = \sum[0.0542087;\ 0.05924638] = 0.11345508$$
$$h_{v1,4}^1 = \sum[0.09507246;\ 0.15143478] = 0.24650724$$
$$h_{v2,4}^1 = \sum[0.09507246;\ 0.15143478;\ 0.21965217;\ 0.1528913] = 0.61905071$$
$$h_{v3,4}^1 = \sum[0.15143478;\ 0.21965217;\ 0.20957971;\ 0.09143478] = 0.67210144$$
$$h_{v4,4}^1 = \sum[0.15143478;\ 0.1528913;\ 0.20957971] = 0.51490579$$
$$h_{v5,4}^1 = \sum[0.21965217;\ 0.1528913;\ 0.20957971;\ 0.19519565] = 0.77731883$$
$$h_{v6,4}^1 = \sum[0.20957971;\ 0.19519565;\ 0.09143478] = 0.49621014$$
$$h_{v7,4}^1 = \sum[0.21965217;\ 0.19519565;\ 0.09143478;\ 0.08042029] = 0.58670389$$
$$h_{v8,4}^1 = \sum[0.09143478;\ 0.08042029] = 0.17185507$$

$$h^1_{v1,5} = \sum[0.07401159;\ 0.14230435] = 0.21631594$$
$$h^1_{v2,5} = \sum[0.07401159;\ 0.14230435;\ 0.15730435;\ 0.13405652] = 0.50767681$$
$$h^1_{v3,5} = \sum[0.14230435;\ 0.15730435;\ 0.17262319;\ 0.04797391] = 0.5202058$$
$$h^1_{v4,5} = \sum[0.14230435;\ 0.13405652;\ 0.17262319] = 0.44898406$$
$$h^1_{v5,5} = \sum[0.15730435;\ 0.13405652;\ 0.17262319;\ 0.14171739] = 0.60570145$$
$$h^1_{v6,5} = \sum[0.17262319;\ 0.14171739;\ 0.04797391] = 0.36231449$$
$$h^1_{v7,5} = \sum[0.15730435;\ 0.14171739;\ 0.04797391;\ 0.05737681] = 0.40437246$$
$$h^1_{v8,5} = \sum[0.04797391;\ 0.05737681] = 0.10535072$$

Therefore, the first iteration produces the following embedding vector:

$$h^1_{vi} = \begin{bmatrix} 0.14757101 & 0.2321884 & 0.21686377 & 0.24650724 & 0.21631594 \\ 0.39402318 & 0.53025362 & 0.52028117 & 0.61905071 & 0.50767681 \\ 0.40090725 & 0.54485508 & 0.56131015 & 0.67210144 & 0.5202058 \\ 0.35413333 & 0.46465942 & 0.46077971 & 0.51490579 & 0.60570145 \\ 0.49152463 & 0.62729 & 0.6276058 & 0.77731883 & 0.44898406 \\ 0.28182029 & 0.3813116 & 0.3783971 & 0.49621014 & 0.60570145 \\ 0.31471984 & 0.42694928 & 0.43175943 & 0.58670389 & 0.40437246 \\ 0.07932754 & 0.11336232 & 0.11345508 & 0.17185507 & 0.10535072 \end{bmatrix}$$

With the $error\ (e)$, it can be calculated as follows:

$$error^l = \frac{\left|\left|h^l_{vi} - h^l_{vj}\right|\right|_{inf}}{|E(G)|^2} \text{ where } i,j \in \{1,2,3,4,5,6,7,8,9\}$$

$$error^0 = \frac{\left|\left|h^1_{v1} - h^1_{v2}\right|\right| + \left|\left|h^1_{v2} - h^1_{v3}\right|\right| + \left|\left|h^1_{v2} - h^1_{v4}\right|\right| + \left|\left|h^1_{v3} - h^1_{v5}\right|\right| + \left|\left|h^1_{v3} - h^1_{v7}\right|\right| +}{|E(G)|^2}$$
$$\frac{\left|\left|h^1_{v4} - h^1_{v5}\right|\right| + \left|\left|h^1_{v5} - h^1_{v6}\right|\right| + \left|\left|h^1_{v6} - h^1_{v7}\right|\right| + \left|\left|h^1_{v7} - h^1_{v8}\right|\right|}{|E(G)|^2}$$

$$error^0 = 0.13804$$

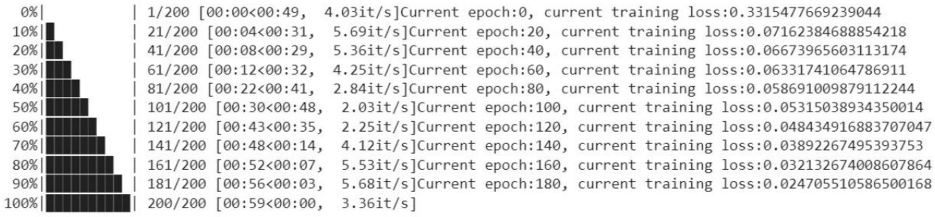
Due to the $error > \epsilon$, we need to update the learning weights W. We update the learning weights using $W^{l+1} = W^l + \alpha \times error^l \times \left(h^l_{vi}\right)^T \times h^{l+1}_{vi}$ until the $error \le \epsilon$. With $\alpha = 0.001$, we can update the weights W as follows:

$$W^{l+1} = W^l + \alpha \times error^l \times \left(h^l_{vi}\right)^T \times h^{l+1}_{vi}$$
$$W^1 = W^0 + \alpha \times error^0 \times \left(h^0_{vi}\right)^T \times h^1_{vi}$$

$$W^l = \begin{bmatrix} 0.02001878 & 0.05002411 & 0.03002702 & 0.10003000 & 0.04002336 \\ 0.01002121 & 0.06002712 & 0.08003034 & 0.02003384 & 0.05002634 \\ 0.03001735 & 0.07002219 & 0.02002499 & 0.09002773 & 0.06002161 \\ 0.04002615 & 0.01003342 & 0.07003737 & 0.05004160 & 0.02003240 \\ 0.08001786 & 0.05002282 & 0.04002577 & 0.03002842 & 0.06002215 \end{bmatrix}$$

Having the new learning weight matrix W^1, we can proceed to the next iteration using the formula $m^{l+1}_{vi} = h^l_{vi} \cdot W^l$, where $l = 0,1,2,3,\cdots,n$ until the $error$ is ≤ 0.001.

Next, a simulation of training, testing, and forecasting traffic violation anomaly data is conducted. The numerical simulation of the graph neural network is performed using Python software.



```
  0%|          | 1/200 [00:00<00:49,  4.03it/s]Current epoch:0, current training loss:0.3315477669239044
 10%|█         | 21/200 [00:04<00:31,  5.69it/s]Current epoch:20, current training loss:0.07162384688854218
 20%|██        | 41/200 [00:08<00:29,  5.36it/s]Current epoch:40, current training loss:0.06673965603113174
 30%|███       | 61/200 [00:12<00:32,  4.25it/s]Current epoch:60, current training loss:0.06331741064786911
 40%|████      | 81/200 [00:22<00:41,  2.84it/s]Current epoch:80, current training loss:0.058691009879112244
 50%|█████     | 101/200 [00:30<00:48,  2.03it/s]Current epoch:100, current training loss:0.05315038934350014
 60%|██████    | 121/200 [00:43<00:35,  2.25it/s]Current epoch:120, current training loss:0.048434916883707047
 70%|███████   | 141/200 [00:48<00:14,  4.12it/s]Current epoch:140, current training loss:0.03892267495393753
 80%|████████  | 161/200 [00:52<00:07,  5.53it/s]Current epoch:160, current training loss:0.032132674008607864
 90%|█████████ | 181/200 [00:56<00:03,  5.68it/s]Current epoch:180, current training loss:0.024705510586500168
100%|██████████| 200/200 [00:59<00:00,  3.36it/s]
```

Figure 6. Loss vs. Epoch

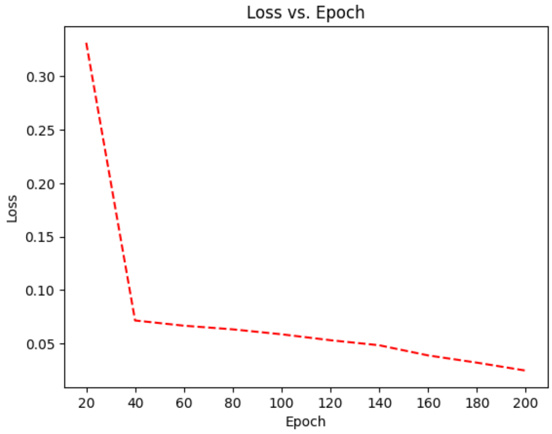


Figure 7. Graphic Loss vs. Epoch

In the training stage illustrated in **Figure 6** and **Figure 7**, we can observe the changes in epoch training loss. Initially, at epoch 0, the training loss has a value of 0.3315, indicating that the initial model struggled to generalize traffic violation patterns. As the training process progresses, the training loss gradually decreases with each epoch. By epoch 20, the training loss decreases to 0.0716, reflecting an improvement in the model's ability to understand and recognize violation patterns. This trend continues, and by epoch 100, the training loss reaches 0.0531, indicating that the model is approaching an accurate representation of traffic violation data. Furthermore, by the end of the training process (epoch 200), the training loss reaches 0.0247, depicting that the model has achieved a high level of precision in predicting and generalizing observed traffic violation patterns. During the 60-day observation period at 8 road points, the testing phase for traffic violations yields a Mean Squared Error (MSE) value of 0.0176. This figure reflects the model's prediction error, with 0.0176 indicating excellent prediction quality.

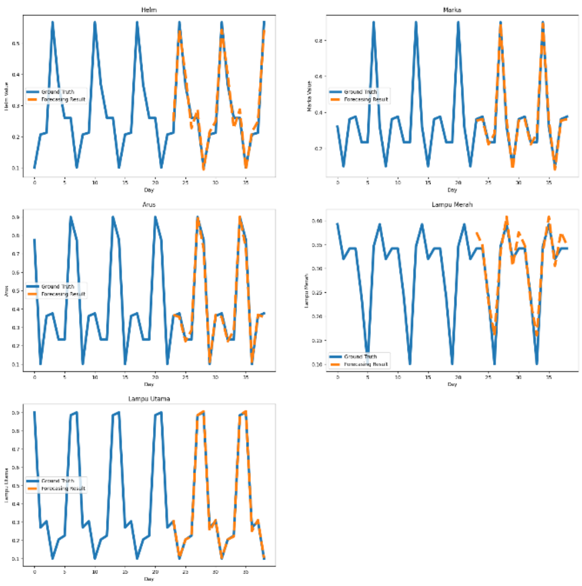


Figure 8. Comparison of Testing and Training Stage Outputs

The results of forecasting traffic violations for the next 14 days have been generated from the analysis of traffic violation data at 8 road points over 60 days. This forecasting process integrates patterns identified by the model during previous training. The forecast results indicate trends and estimated traffic violations that may occur at each road point during the upcoming 14-day period. This process provides valuable insights for anticipating and managing potential traffic violations, assisting authorities in taking more effective preventive measures. The 14-day forecast can be observed in **Figure 9**.

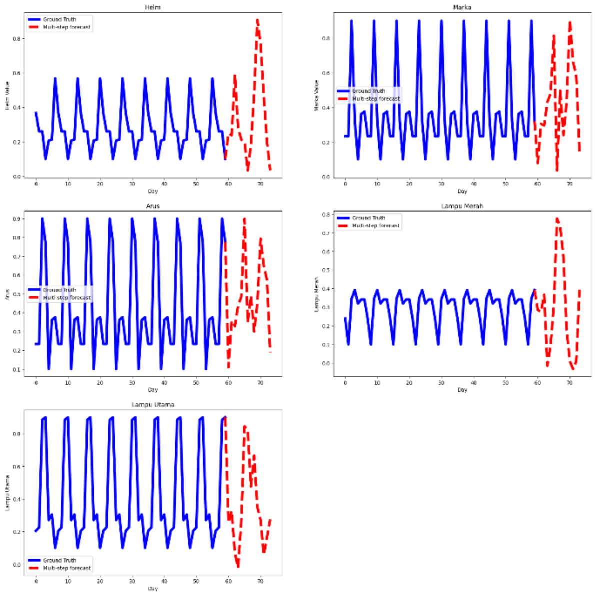


Figure 9. The Results of The 14-Day Forecast

4. CONCLUSIONS

In our paper, we analyze the resolving efficient domination number of several graphs including $P_n \odot P_2$, $S_n \odot P_2$, $C_n \odot P_2$, and $Shack(G, u_i, n)$. Based on this research, it is found that $\gamma_{re}\ (P_n \odot P_2\ ) = n$, $\gamma_{re}\ (S_n \odot P_2\ ) = n + 1$, $\gamma_{re}\ (C_n \odot P_2\ ) = n$, and $\gamma_{re}\ (Shack(G, u_i, n)) = 2n + 1$. The resulting theorem aims to determine the number of traffic violation sensors required within a road map. Subsequently, traffic violation simulation data is collected from these road points to forecast violations. The STGNN algorithm is divided into three stages: training, testing, and forecasting. The generated outputs indicate that the final model performs exceptionally well in reducing prediction errors and producing estimates that closely approximate the original data with high accuracy. During the 60-day observation period at 8 road points, the testing phase for traffic violations resulted in a Mean Squared Error (MSE) value of 0.0176. This figure reflects the model's prediction error, with 0.0176 indicating excellent prediction quality. This process provides valuable insights into anticipating and managing potential traffic violations, as well as assisting authorities in taking more effective preventive measures.

Based on the research on resolving efficient dominating set in determining the value of the resolving efficient domination number on $P_n \odot P_2$, the graph $S_n \odot P_2$, the graph $C_n \odot P_2$, and the $Shack(G, u_i, n)$ graph, the researchers suggest other researchers conduct studies on resolving efficient dominating set in graphs resulting from other operations. Additionally, the researchers also recommend other researchers to apply resolving efficient dominating set in other application schemes.

ACKNOWLEDGMENT

The authors would like to express their gratitude to CGANT Research Group of Universitas Jember for their support.

REFERENCES

[1] F. A. Abdullah and F. Windiyastuti, "Electronic Traffic Law Enforcement (ETLE) sebagai Digitalisasi Proses Tilang," *J. Kewarganegaraan*, vol. 6, no. 2, pp. 3004–3008, 2022.

[2] V. M. Benyamin Lufpi, "Efektivitas Electronic Traffic Law Enforcement," *J. Ilmu Kepol.*, vol. 16, no. 1, p. 9, 2022, doi: 10.35879/jik.v16i1.350.

[3] R. Diestel, *Graph Theory (on Demand Printing Of 02787)*. 2018. doi: 10.1201/9780429493768.

[4] B. G. Theory and O. Riordan, "B8.511b Graph Theory," 2016.

[5] R. A. Hakim, Dafik, I. M. Tirta, R. M. Prihandini, and I. H. Agustin, "On resolving efficient domination number of comb product of special graphs," *J. Phys. Conf. Ser.*, vol. 1832, no. 1, 2021, doi: 10.1088/1742-6596/1832/1/012018.

[6] I. Kusumawardani, Dafik, E. Y. Kurniawati, I. H. Agustin, and R. Alfarisi, "On resolving efficient domination number of path and comb product of special graph," *J. Phys. Conf. Ser.*, vol. 2157, no. 1, 2022, doi: 10.1088/1742-6596/2157/1/012012.

[7] Y. Wangguway, Slamin, D. Dafik, D. A. R. Wardani, and R. Alfarisi, "On resolving domination number of special family of graphs," *J. Phys. Conf. Ser.*, vol. 1465, no. 1, 2020, doi: 10.1088/1742-6596/1465/1/012015.

[8] A. Brandstädt, A. Leitert, and D. Rautenbach, "Efficient dominating and edge dominating sets for graphs and hypergraphs," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7676 LNCS, no. July 2012, pp. 267–277, 2012, doi: 10.1007/978-3-642-35261-4_30.

[9] Dafik, I. H. Agustin, E. R. Albirri, R. Alfarisi, and R. M. Prihandini, "Locating domination number of m-shadowing of graphs," *J. Phys. Conf. Ser.*, vol. 1008, no. 1, 2018, doi: 10.1088/1742-6596/1008/1/012041.

[10] R. C. Brigham, G. Chartrand, R. D. Dutton, and P. Zhang, "Resolving domination in graphs," *Math. Bohem.*, vol. 128, no. 1, pp. 25–36, 2003, doi: 10.21136/mb.2003.133935.

[11] G. Chartrand, F. Okamoto, and P. Zhang, "The metric chromatic number of a graph," *Australas. J. Comb.*, vol. 44, no. 2, pp. 273–286, 2009.

[12] D. A. R. Wardani, D. Dafik, I. H. Agustin, M. Marsidi, and C. D. Putri, "On the LIDS of corona product of graphs," *AIP Conf. Proc.*, vol. 2014, pp. 1–8, 2018, doi: 10.1063/1.5054491.

[13] R. Adawiyah and R. M. Prihandini, "Dimensi Metrik Sisi Pada Keluarga Graf Tangga," pp. 34–39, 2021.

[14] R. Alfarisi, Dafik, R. Adawiyah, R. M. Prihandini, E. R. Albirri, and I. H. Agustin, "Super domination number of unicyclic graphs," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 243, no. 1, 2019, doi: 10.1088/1755-1315/243/1/012074.

[15] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009, doi: 10.1109/TNN.2008.2005605.