Global Journal of Engineering and Technology [GJET].
ISSN: 2583-3359 (Online)
Frequency: Monthly
Published By GSAR Publishers
Journal Homepage Link- https://gsarpublishers.com/journal-gjet-home/

A Distinct Weighted Centrality–RMS Graph Labeling for Cybersecurity Network Prioritization

By

Sreena T.D.
Department of Mathematics, Sree Narayana College, Nattika, Thrissur-680566, Kerala, India

Article History

Received: 25/07/2026
Accepted: 02/08/2026
Published: 05/08/2026

Vol – 5 Issue – 8

PP: - 13-17

Abstract

Graph labeling is a useful way to study the structure of complex networks. In this study, Distinct Weighted Centrality–Root Mean Square Graph Labeling (DWCRMSGL), a method that combines betweenness, degree, and closeness centralities to give unique weighted labels to each vertex is presented. Edge labels using the Root Mean Square (RMS) of the labels of their connected vertices are computed. Also describes an algorithm for this labeling and explain its theoretical properties. Later test the method on the Email-Eu-Core communication network, using the vertex and edge labels to find and rank key users and communication links. This approach allows us to bring together several centrality measures in graph labeling, making it easier to identify important parts of a network. In cybersecurity, this method can help with network monitoring, resource allocation, and assessing communication risks. The labeling is efficient and can be applied to other real-world communication and complex networks.

Keywords: *Graph labeling, Network centrality, Betweenness centrality, Root Mean Square (RMS), Email-Eu-Core network, Cybersecurity.*

1. INTRODUCTION

Graph theory is a useful mathematical tool for modeling and analyzing complex systems, where entities are shown as vertices and their interactions as edges. Graph labeling, a key area within graph theory, assigns labels to vertices, edges, or both based on specific rules. This area has gained attention for its theoretical value and practical uses in fields like communication networks, transportation, coding theory, and computer science [2–4,8]. As communication networks become more complex, there is a growing need for graph labeling methods that can clearly show the structural importance of different network parts.

Graph-theoretic measures help analyze complex communication networks by measuring how much influence each vertex has. Centrality measures like degree, betweenness, and closeness look at different sides of vertex importance, such as how well a vertex connects locally, how much it controls information flow, and how efficiently it communicates. These measures are often used in network optimization, communication analysis, and cybersecurity to find key nodes and improve how networks are managed [1,5,6,9–11].

Over the years, many graph labeling techniques have been developed to study different types of graphs and their structures. Gallian [4] provided a thorough review of these methods, showing how quickly the field is growing and how widely it is used. Standard graph theory books by Chartrand and Lesniak [2], West [8], and Diestel [3] also cover many labeling techniques and their mathematical details. At the same time, graph-based methods have been used for analyzing communication networks, routing, and checking for weaknesses [1,5,11]. In cybersecurity, graph models help study intrusion detection systems, network vulnerabilities, and communication security [6,7,11].

Although there is a lot of research on graph labeling and network analysis, most labeling methods rely on combinatorial rules and do not use structural information from centrality measures. Similarly, centrality measures are usually just used to rank vertices and do not offer a single labeling system that gives unique labels to both vertices and edges. Also, edge importance is often measured on its own, without looking at the combined importance of the vertices it connects. The proposed labeling assigns distinct weighted labels to vertices using a weighted combination of betweenness, degree, and closeness centralities, while edge labels are computed using the Root Mean Square (RMS) of the labels of their incident vertices. An algorithm for constructing the proposed labeling is presented, and its mathematical properties are established. The effectiveness of the proposed method is demonstrated using the Email-Eu-Core communication network, where the computed labels are used to identify and rank critical users and communication links for cybersecurity applications.

*Corresponding Author: Sreena T.D

2. MAIN RESULTS

Definition 2.1

Let G = (V, E) be a simple graph where $|V|$= n. Let B(v)=normalized betweenness centrality,
D(v)= normalized degree centrality, and C (v) = normalized closeness centrality. Choose real constants w_1, w_2, w_3 satisfying

- $w_1 + w_2 + w_3 = 1$,
- $w_1 > w_2 > w_3 > 0$.

The vertex labeling function $f: V \to [0,1]$ is defined by $f(v) = w_1\mathrm{B(v)} + w_2\mathrm{D(v)} + w_3\mathrm{C(v)}$.

The induced edge labeling function g : E → [0,1] is defined by $\mathrm{g}(u,v) = \sqrt{\frac{f(u)^2+f(v)^2}{2}}$.The graph G is called a Weighted Centrality RMS Graph.

If both f and g are injective, then G is called a Distinct Weighted Centrality RMS Graph.

Example 2.2

Consider path graph P_4 with four vertices v_1, v_2, v_3, and v_4 .Assume $w_1 = 0.6$, $w_2 = 0.25$, $w_3 = 0.15$.

Suppose the normalized centralities are as in Table 1.

Vertex	B(v)	D(v)	C(v)	f(v)
v1	0	0.33	0.50	0.16
v2	0.67	0.67	0.75	0.68
v3	0.67	0.67	0.75	0.68
v4	0	0.33	0.50	0.16

Table 1: Computation of f

Edge	(v1,v2)	(v2,v3)	(v3,v4)
g	0.35	0.48	0.35

Table 2: Computation of g

Figure 1 illustrates this example :

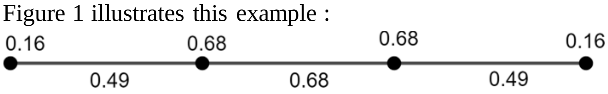

Fig.1: An example of Weighted Centrality RMS Graph

Arrange the vertices in the ascending order of vertex label as in Table 3.

Vertex	f(v)	Rank	Identifier(New vertex label) f '(v)
v1	0.16	1	0.10
v4	0.16	2	0.20
v2	0.68	3	0.30
v3	0.68	4	0.40

Table 3: Ranking of vertices

In order to avoid tie, compare the different centralities of v_1 , v_4 and v_2 , v_3 separately.

Redefine $g(u,v) = \sqrt{\frac{rank(u)f(u)^2+rank(v)f(v)^2}{2}}$

Centralities remaining the same. Since still tied, compare their vertex identifiers

Edge	(v1,v2)	(v2,v3)	(v3,v4)
g'(u,v)	0.16	0.25	0.35

Table 4: Edge labels induced by the reordered weighted vertex labeling.

The new values make P4 a Distinct Weighted Centrality RMS Graph.

Computation Procedure 2.3 : Construction of a Distinct Weighted Centrality RMS Graph

Input: A connected simple graph $G = (V, E)$
Output: A Distinct Weighted Centrality RMS Graph

1. Compute the normalized betweenness, degree, and closeness centralities for each vertex.
2. Choose weights w_1, w_2, w_3 such that $w_1 + w_2 + w_3 = 1$, $w_1 > w_2 > w_3 > 0$.
3. Compute the composite vertex score $f(v) = w_1B(v) + w_2D(v) + w_3C(v)$.
4. Resolve any ties in the vertex scores using a predetermined tie-breaking rule to obtain distinct vertex labels.
5. For each edge (u, v), compute the edge label $g(u,v) = \sqrt{\frac{f(u)^2+f(v)^2}{2}}$.
6. If both the vertex and edge labels are distinct, then G is a Distinct Weighted Centrality RMS Graph; otherwise, G does not admit the proposed labeling for the chosen weights.

Theorem 2.4

Let $e = (u,v)$ be an edge of a Weighted Centrality RMS Graph. Then
$\min\{f(u), f(v)\} \le g(e) \le \max\{f(u), f(v)\}$.

Proof

By definition, $g(e) = \sqrt{\frac{f(u)^2+f(v)^2}{2}}$.
Assume, without loss of generality, that $f(u) \le f(v)$.
Then $f(u)^2 \le \frac{f(u)^2+f(v)^2}{2} \le f(v)^2$.
Taking square roots ,$f(u) \le g(e) \le f(v)$.
Hence,$\min\{f(u), f(v)\} \le g(e) \le \max\{f(u), f(v)\}$.
Thus the theorem is proved.

Theorem 2.5

Suppose $f(u) \le f(v) \le f(w)$. Then $g(u,v) \le g(u,w)$.

Proof

Since $g(u,v) = \sqrt{\frac{f(u)^2+f(v)^2}{2}}$, and $g(u,w) = \sqrt{\frac{f(u)^2+f(w)^2}{2}}$, we have $f(v)^2 \le f(w)^2$.

*Corresponding Author: Sreena T.D

Therefore, $\frac{f(u)^2+f(v)^2}{2} \leq \frac{f(u)^2+f(w)^2}{2}$.
Taking square roots gives $g(u,v) \leq g(u,w)$.
Hence proved.

Theorem 2.6
If two distinct edges $e_1 = (u,v)$, $e_2 = (x,y)$ satisfy $f(u)^2 + f(v)^2 = f(x)^2 + f(y)^2$,
then $g(e_1) = g(e_2)$.

Proof

By definition, $g(e_1) = \sqrt{\frac{f(u)^2+f(v)^2}{2}}$ and $g(e_2) = \sqrt{\frac{f(x)^2+f(y)^2}{2}}$.
Since $f(u)^2 + f(v)^2 = f(x)^2 + f(y)^2$, the quantities under the square roots are equal. Hence, $g(e_1) = g(e_2)$.

Therefore, if two different edges have equal sums of squared endpoint labels, they have identical RMS labels.

Theorem 2.7
If every pair of distinct edges has different sums of squared endpoint labels, then the RMS edge labeling is injective.

Proof
Suppose $e_1 \neq e_2$.
By hypothesis, $f(u)^2 + f(v)^2 \neq f(x)^2 + f(y)^2$.
Hence, $\frac{f(u)^2+f(v)^2}{2} \neq \frac{f(x)^2+f(y)^2}{2}$.
Taking square roots, $g(e_1) \neq g(e_2)$.Thus every edge has a distinct RMS label.
Therefore g is injective.

3. Experimental Validation Using the Email-Eu-Core Dataset

The proposed Distinct Weighted Centrality RMS Labeling is validated using the Email-Eu-Core network dataset. The network consists of employees of a European research institution, where each vertex represents an employee and each edge represents an email communication between two employees.

3.1 Methodology

The following procedure is adopted.
Step 1:Construct the graph $G = (V, E)$ from the Email-Eu-Core dataset.

Step 2:Compute Betweenness Centrality, Degree Centrality and Closeness Centrality for every vertex.
Step 3:Assign the vertex labels using $f(v) = w_1$ B(v) + w_2 D(v) + w_3 C(v).

Step 4:Compute the edge labels $g(u\cdot v) = \sqrt{\frac{f(u)^2 + f(v)^2}{2}}$.
Step 5:Rank all vertices according to the proposed labels.

3.2 Python Implementatio

The Email-Eu-Core communication network was analyzed using the Python programming language. The network was constructed from the dataset using the NetworkX library, and the degree, betweenness, and closeness centralities of all vertices were computed. Based on these centrality values, the proposed weighted vertex labels and RMS edge labels were calculated according to the proposed labeling scheme. The resulting vertex and edge labels were ranked and exported using the Pandas library for further analysis and visualization. The generated CSV files were subsequently imported into Gephi to visualize the network topology.

```
import networkx as nx
import pandas as pd
from google.colab import files
# Upload dataset

Uploaded files.upload() filename = list(uploaded.keys())[0] # Read graph

G = nx.read_edgelist (filename, create_using=nx.DiGraph(), nodetype=int) # Centralities

dc = nx.degree_centrality(G)

bc = nx.betweenness_centrality(G, normalized=True) cc = nx.closeness_centrality(G)

# Weights

w1, w2, w3 = 0.5, 0.3, 0.2

# Vertex labels vlabel = {v: w1*bc[v] + w2*dc[v] + w3*cc[v] for v in G.nodes()} # Edge labels (RMS)

elabel = {(u, v): ((vlabel[u]**2 + vlabel[v]**2)/2)**0.5

for u, v in G.edges()} # Vertex tablevertex_df = pd.DataFrame({ "Vertex": list(G.nodes()),

"Degree": [dc[v] for v in G.nodes()],

"Betweenness": [bc[v] for v in G.nodes()], "Closeness": [cc[v] for v in G.nodes()], "Weighted Label": [vlabel[v] for v in G.nodes()]

}).sort_values("Weighted Label", ascending=False) vertex_df["Rank"] = range(1, len(vertex_df)+1)

# Edge table edge_df = pd.DataFrame(

[(u, v, elabel[(u, v)]) for u, v in G.edges()], columns=["Source", "Target", "RMS Edge Label"]

).sort_values("RMS Edge Label", ascending=False) edge_df["Rank"] = range(1, len(edge_df)+1)

# Save files vertex_df.to_csv("Vertex_Centralities.csv", index=False)edge_df.to_csv("Edge_Labels.csv", index=False)withpd.ExcelWriter("Weighted_Centrality_RMS_Labeling.xlsx") as writer: vertex_df.to_excel(writer, sheet_name="Vertices", index=False) edge_df.to_excel(writer, sheet_name="Edges", index=False)

# Download files.download("Vertex_Centralities.csv") files.download("Edge_Labels.csv")

files.download("Weighted_Centrality_RMS_Labeling.xlsx")
```

3.3 Network Visualization

To illustrate the labeling, the Email-Eu-Core communication network was visualized using Gephi. Figure 2 illustrates the communication topology of the Email-Eu-Core network used

*Corresponding Author: Sreena T.D

for validating the proposed labellingThe vertex labels correspond to the ranking obtained from the weighted centrality values, and the edge labels are assigned according to the proposed RMS labeling scheme.

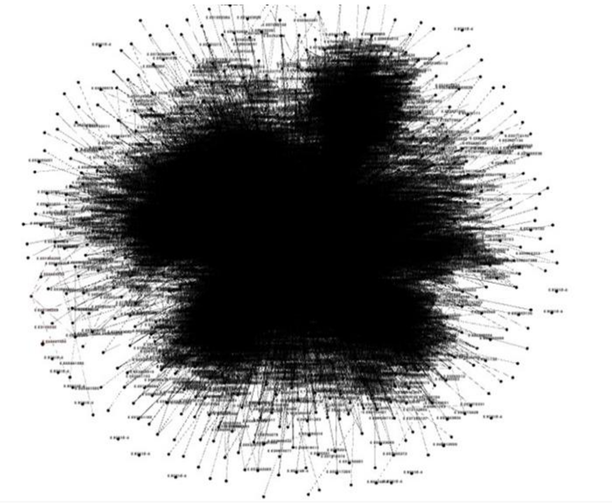

Figure 2: Email-Eu-Core Communication Network

3.4 Highest Ranked Vertices

The computed weighted vertex labels were arranged in descending order to obtain the vertex ranking. The highest-ranked vertices are identified as the most critical users in the Email-Eu-Core network, indicating their significant influence on communication flow and their priority for cybersecurity monitoring and protection.

3.5 Highest Ranked Edges

The RMS edge labels were computed for all communication links and ranked in descending order. Edges with the highest RMS values represent the most critical communication channels connecting influential users and should be prioritized for secure communication, traffic monitoring, and intrusion detection.

3.6 Cybersecurity Interpretation

Interpret the results rather than merely listing them.

- High-ranked vertices correspond to users that combine strong connectivity, efficient communication, and significant bridging capability.
- Compromise of such vertices may facilitate rapid dissemination of malware, phishing campaigns, or unauthorized information flow.
- These vertices should therefore receive priority for multifactor authentication,
- continuous monitoring, vulnerability assessment, and timely patch management.
- High-ranked edges represent communication channels between influential users. Monitoring these links enables early detection of suspicious communication patterns and improves resource allocation for intrusion detection.

3.7 Advantages of the Proposed Labeling

1. It combines three complementary centrality measures into a single distinct vertex label.
2. RMS-based edge labeling captures the joint importance of connected vertices.
3. The ranking provides a practical prioritization of nodes and links for cybersecurity operations.
4. The method is applicable to organizational email, social, transportation, and communication networks.

4. CONCLUSION

This paper introduced a Distinct Weighted Centrality–Root Mean Square Graph Labeling method. It assigns unique weighted labels to vertices based on betweenness, degree, and closeness centralities. Edge labels are calculated using the Root Mean Square of the connected vertex labels. The paper presented an algorithm for this labeling and discussed its theoretical properties. The method was tested on the Email-Eu-Core communication network, where the labels helped identify and rank key users and communication links. The results show that this labeling offers a useful way to prioritize important parts of a network based on their structure. Because it is simple and efficient, this approach could be used in cybersecurity, communication, social, and other complex networks. However, the method only uses structural centrality measures and does not account for changes over time, such as shifting communication patterns or new attack strategies. As a result, the labels show the structural importance of vertices and edges, not real-time network activity. Future research could adapt this method for dynamic networks and explore ways to select weights automatically.

REFERENCES

1. Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network flows: Theory, algorithms, and applications*. Prentice Hall.
2. Chartrand, G., & Lesniak, L. (1996). *Graphs & digraphs*. CRC Press.
3. Diestel, R. (2017). *Graph theory* (5th ed.). Springer.
4. Gallian, J. A. (2009). A dynamic survey of graph labeling. *The Electronic Journal of Combinatorics.*
5. Kleinberg, J., & Tardos, É. (2006). *Algorithm design*. Pearson.
6. Scarfone, K., & Mell, P. (2007). *Guide to intrusion detection and prevention systems (IDPS),*NIST Special Publication.
7. Smith, R. E. (2019). *Elementary information security*. Jones & Bartlett Learning.
8. West, D. B. (2001). *Introduction to graph theory* (2nd ed.). Prentice Hall.
9. Forouzan, B. A. (2007). *Data communications and networking*. Huga Media.
10. Kurose, J. F., & Ross, K. W. (2012). Computer networking: A top-down approach sixth edition. *Citado na*, *5*.

*Corresponding Author: Sreena T.D

11. Ammann, P., Wijesekera, D., & Kaushik, S. (2002, November). *Scalable, graph-based network vulnerability analysis.* In Proceedings of the 9th ACM Conference on Computer and Communications Security (pp. 217-224).

*Corresponding Author: Sreena T.D