Acta Universitatis Sapientiae

Informatica

Volume 15, Number 1, 2023

Sapientia Hungarian University of Transylvania
Scientia Publishing House

Journal Metrics (2022)

Impact Factor: **0.3**
Five Year Impact Factor: **0.8**
JCI: **0.09**
MEiN: **20**
Google Scholar h5-index: **21**
Google Scholar h5-median: **28**

Acta Universitatis Sapientiae Informatica
is covered by the following services:

ACM Digital Library
Baidu Scholar
Cabell's Journalytics
CNKI Scholar
CNPIEC – cnpLINKer Dimensions
DOAJ
EBSCO
Engineering Village
ExLibris
Google Scholar
Inspec
Japan Science and Technology Agency
J-Gate
JournalTOCs
KESLI-NDSL
MyScienceWork
Naver Academic
Naviga (Softweco)
QOAM
ReadCube
SCILIT
Semantic Scholar
Sherpa/RoMEO
TDNet
Ulrich's Periodicals Directory
WanFang Data
Web of Science – ESCII
WorldCat (OCLC)
zbMATH Open
X-MOL

Contents

Acta Univ. Sapientiae Informatica **15,** 1 (2023) 1–9

DOI: 10.2478/ausi-2023-0001

On domination in signed graphs

James JOSEPH
CHRIST(Deemed to be University)
Bangalore
Karnataka, India
email:
james.joseph@res.christuniversity.in

Mayamma JOSEPH
CHRIST(Deemed to be University)
Bangalore
Karnataka, India
email:
mayamma.joseph@christuniversity.in
ORCID:0000-0001-5819-247X

Abstract. In this article the concept of domination in signed graphs is examined from an alternate perspective and a new definition of the same is introduced. A vertex subset D of a signed graph S is a dominating set, if for each vertex v not in D there exists a vertex $u \in D$ such that the sign of the edge uv is positive. The domination number $\gamma(S)$ of S is the minimum cardinality among all the dominating sets of S. We obtain certain bounds of $\gamma(S)$ and present a necessary and sufficient condition for a dominating set to be a minimal dominating set. Further, we characterise the signed graphs having small and large values for domination number.

1 Introduction

A *signed graph* is an ordered pair $S = (G, \sigma)$, where $G = (V, E)$ is a simple graph called the *underlying graph* of S and $\sigma : E(G) \to \{-1, 1\}$ is a function called a *signing* of G or the *signature* of S. The negative and positive edges are depicted using dashed and solid lines respectively. The set of all positive(negative) edges is denoted by $E^+(S)(E^-(S))$. The subgraph obtained by removing the negative(positive) edges is denoted by $S^+(S^-)$. A signed

Key words and phrases: signed graphs, dominating set, domination number, minimal dominating set

graph where every edge is positive(negative) is called *all-positive(all-negative)*. For any vertex u, $N^+(u) = \{v \in N(u) | \sigma(uv) = 1\}$ and $N^-(u) = \{v \in N(u) | \sigma(uv) = -1\}$. The positive and negative degree of a vertex u is defined as $d^+(u) = |N^+(u)|$ and $d^-(u) = |N^-(u)|$ respectively, while the degree of u in G is $d_G(u) = d^+(u) + d^-(u)$. The maximum(minimum) positive degree of S is denoted using $\Delta^+(S)$ ($\delta^+(S)$), whereas $\Delta^-(S)$ ($\delta^-(S)$) denote the maximum (minimum) negative degree of S. Signed graphs provide a large scope for researchers to investigate both theoretical and application problems in graph theory [1, 2, 5, 7, 13, 15, 17, 18, 19].

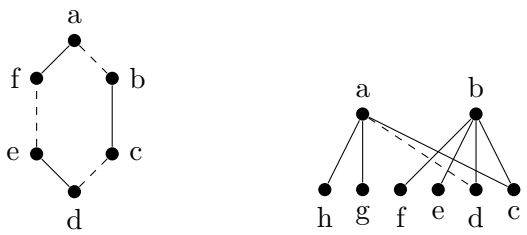
The concept of domination in graphs is a well established research area in graph theory [6, 8, 9, 10, 11, 12]. Although the concept of domination in graphs was introduced by Berge [4] in the year 1962, it is to be noted that the first article on domination in signed graphs appeared only in the year 2013. Another interesting fact is that the idea of domination can be viewed from various perspectives. It was Acharya[2] who made the first attempt in articulating the concept of domination in signed graphs. He defined a dominating set of a signed graph $S = (G, \sigma)$ as a set $D \subseteq V$ such that all the vertices of S are either in D or there exists a function $\mu : V \to \{-1, 1\}$ called a marking of S such that all the vertices $u \in V \setminus D$ are adjacent to at least one vertex $v \in D$ such that $\sigma(uv) = \mu(u)\mu(v)$. Later in the year 2020, Jeyalakshmi [13] proposed another definition for a dominating set of a signed graph. A subset D of the vertex set V is called a dominating set of a signed graph S if for all $v \in V \setminus D$, $|N^+(v) \cap D| > |N^-(v) \cap D|$. In this article we study the concept of domination in signed graphs from yet another point of view.

Joseph and Joseph [14] considered the fact that in any network that can be represented as a signed graph, a vertex dominates another vertex provided there exist a positive edge between them. In this sense, a set D of vertices of a signed graph S that are connected to the remaining vertices of S by positive edges can be considered as a dominating set of S. Accordingly they presented the following definition for a dominating set of signed graphs.

Definition 1 [14] *Let $S = (G, \sigma)$ be a signed graph. A set $D \subseteq V$ is said to be a dominating set of S if each vertex $v \in V \setminus D$ is adjacent to at least one vertex $u \in D$ such that $\sigma(uv) = 1$. The minimum cardinality among all the dominating sets of S is called the domination number of S, denoted by $\gamma(S)$.*

For a signed graph $S = (G, \sigma)$, by the term $\gamma(S)$-set we mean a minimum dominating set and $\gamma(G)$-set refers to a minimum dominating set of the underlying graph G.

Clearly, if all the edges of S are positive then the above definition reduces to that of the domination in graphs. On the other hand, if S is all negative, then the dominating set is trivially V.



(a) (b)

Figure 1

If $|N^+(v) \cap D| > |N^-(v) \cap D|$ $\forall v \in V \setminus D$, then we can observe that $N^+(v) \cap D$ is non-empty whenever $v \in V \setminus D$ i.e every vertex $v \in V \setminus D$ is adjacent to a vertex $u \in D$ such that $uv \in E^+(S)$. Hence, any dominating set as given in [13] is also a dominating set as per Definition 1. For example, the signed graph in Figure 1(a) has $\{a, b, c, d\}$ as a minimum dominating set by the definition proposed by Jeyalakshmi where as $\{a, b, d\}$ is a minimum dominating set by Definition 1. But, for the signed graph given in Figure 1(b), $\{a, b\}$ is the minimum dominating set in both the cases.

2 Preliminaries

In this section we explore some of the basic results that follow from Definition 1. A characterisation of minimal dominating sets in graphs was obtained by Ore [16]. Analogously we present a characterisation for the minimal dominating sets of a signed graph. We omit the proof as it is similar to the corresponding result in [16].

Theorem 2 *Let S be a signed graph and D be a dominating set of S. Then D is a minimal dominating set of S if and only if for each vertex v in D either of the following conditions hold:*

(i) $N^+(v) \subseteq V \setminus D$,

(ii) There exists a vertex u in $V \setminus D$ such that $N^+(u) \cap D = \{v\}$.

For any signed graph S of order n, the vertex set is a trivial dominating set. On the other hand any dominating set of S is of cardinality at least 1 so that we have the obvious inequality $1 \leq \gamma(S) \leq n$. The upper bound is attained by all-negative signed graphs. Moreover, for any signed graph S of order n, $\gamma(S) = n$ if and only if S is all-negative.

Remark 3 *In any signed graph* S, *a vertex* v *with* $N^+(v) = \phi$ *belongs to all the dominating sets of* S.

Note that the dominating sets of signed graphs are always the dominating sets of their underlying graphs. Thus we have the following inequality.

Proposition 4 *If* $S = (G, \sigma)$ *is any signed graph, then* $\gamma(S) \geq \gamma(G)$.

The following theorem characterises the signed graphs having domination number equal to that of their underlying graphs.

Theorem 5 *Let* $S = (G, \sigma)$ *be a signed graph. Then* $\gamma(S) = \gamma(G)$ *if and only if there is a* $\gamma(G)$*-set* D *such that for very vertex* u *in* $V \setminus D$, $N^+(u) \cap D \neq \phi$.

Proof. Suppose that $S = (G, \sigma)$ is a signed graph such that $\gamma(S) = \gamma(G)$. Then there is a $\gamma(S)$-set D which is a $\gamma(G)$-set. Also, by the definition of a dominating set of a signed graph, for every vertex u in $V \setminus D$, $N^+(u) \cap D \neq \phi$.

Conversely, suppose that there exists a $\gamma(G)$-set D such that for every vertex u in $V \setminus D$, $N^+(u) \cap D \neq \phi$. Therefore D is a dominating set of S and $\gamma(S) \leq \gamma(G)$. Then from Proposition 4 it follows that $\gamma(S) = \gamma(G)$. □

Now we examine the number of negative edges in a signed graph $S = (G, \sigma)$ such that $\gamma(S) \geq \gamma(G)$. Recall that the bondage number $b(G)$ of a graph G is defined as the minimum number of edges whose removal increases the domination number. Let $m^-(S)$ be the number of negative edges in S. Then we have the following proposition.

Proposition 6 *Let* $S = (G, \sigma)$ *be a signed graph.*

(i) If $\gamma(S) = \gamma(G)$, *then the maximum value of* $m^-(S)$ *is the number of edges between the vertices of a* $\gamma(G)$*-set, say* D *and the number of edges between the vertices of the set* $V \setminus D$ *and* $|N(v) \cap D| - 1$ *edges corresponding to each vertex* v *in* $V \setminus D$ *dominated by more than one vertex in* D.

(ii) If $\gamma(S) > \gamma(G)$, *then* $m^-(S) \geq b(G)$.

Proposition 7 follows from the fact that joining any two non-adjacent vertices of a signed graph by a negative edge does not change the domination number.

Proposition 7 *Let S be any signed graph with a pair of non-adjacent vertices and S' be the signed graph obtained from S by joining a pair non-adjacent vertices of S by a negative edge, then $\gamma(S) = \gamma(S')$.*

Remark 8 *From Proposition 7 it follows that $\gamma(S) = \gamma(S^+)$.*

We use Ore's theorem to obtain a bound for domination number of certain class of signed graphs.

Theorem 9 (Ore' s Theorem[16]) *If a graph G has no isolated vertices, then $\gamma(G) \leq \frac{n}{2}$.*

The following result gives a bound similar to the bound given in Ore's Theorem on the domination number of signed graphs S with positive $\delta^+(S)$.

Theorem 10 *If S is any signed graph of order n such that $\delta^+(S) > 0$, then $\gamma(S) \leq \frac{n}{2}$.*

Proof. Given that S is any signed graph with $\delta^+(S) > 0$, then $d^+(v) > 0$ for every vertex $v \in V$. This implies that the subgraph S^+ is a graph without any isolates. Therefore by Ore's Theorem $\gamma(S^+) \leq \frac{n}{2}$. Now by applying Remark 8, $\gamma(S) \leq \frac{n}{2}$. □

The corona of two graphs G and H, denoted by $G \circ H$, is the graph obtained by taking one copy of G and $|V(G)|$ copies of H such that i^{th} vertex of G is adjacent to all the vertices of i^{th} copy of H. We refer the following theorem to obtain a characterisation of the signed graphs with order n having domination number $\frac{n}{2}$, when n is even.

Theorem 11 [11] *For any graph G with even order n and no isolated vertices, $\gamma(G) = n/2$ if and only if the components of G are the cycle C_4 or the corona $H \circ K_1$, where H is any connected graph.*

Analogously, we have the next result that follows from Theorem 11 and Remark 8.

Theorem 12 *For any signed graph S of even order n and $\delta^+(S) > 0$, $\gamma(S) = n/2$ if and only if the components of S^+ are the cycle C_4 or the corona $H \circ K_1$, where H is any connected graph.*

Haynes et al. [3] have defined a class of graphs $\mathcal{G}$ inorder to characterise the connected graphs G with $\gamma(G) = \lfloor \frac{n}{2} \rfloor$ and the characterisation is as follows.

Theorem 13 [3] *A connected graph* G *have* $\gamma(G) = \lfloor \frac{n}{2} \rfloor$ *if and only if* $G \in \mathcal{G}$.

In this direction, using Remark 8 and Theorem 13 we have the corresponding result for signed graphs S with the property that S^+ is connected.

Theorem 14 *For any signed graph* S *that contains a connected* S^+, $\gamma(S) = \lfloor \frac{n}{2} \rfloor$ *if and only if* $S^+ \in \mathcal{G}$.

3 Main results

Now we examine the signed graphs with small and large values for domination number. First we obtain the characterisation for signed graphs S with $\gamma(S) = 1$. Whenever S is a signed graph with $\gamma(S) = 1$ and $\{v\}$ is a $\gamma(S)$-set, then v is a vertex with $d^+(v) = n - 1$, where n is the order of S. This implies that $\Delta^+(S) = n-1$. Conversely, if $\Delta^+(S) = n-1$, then $\gamma(S) = 1$. Thus we have the following characterisation for signed graphs with domination number equal to 1.

Theorem 15 *For any signed graph* S *of order* n, $\gamma(S) = 1$ *if and only if* $\Delta^+(S) = n - 1$.

From the above theorem we can see that for any signed graph S of order n with $\gamma(S) > 1$, $\Delta^+(S) \leq n - 2$. The following theorem characterises signed graphs with domination number equal to 2.

Theorem 16 *Let* S *be any signed graph of order* n. *Then* $\gamma(S) = 2$ *if and only if* $\Delta^+(S) \leq n - 2$ *and there exists a pair of vertices* u *and* v *such that* $V \setminus \{u, v\} \subseteq N^+(u) \cup N^+(v)$.

Proof. Assume that $\gamma(S) = 2$ and let $D = \{u, v\}$ be a $\gamma(S)$-set of S. By Theorem 15, $\Delta^+(S) \leq n-2$. Since each vertex in $V \setminus D$ is adjacent to at least one vertex in D by a positive edge, $V \setminus D \subseteq N^+(u) \cup N^+(v)$.

Suppose that $\Delta^+(S) \leq n-2$ and there exist a pair of vertices u and v such that$V \setminus \{u, v\} \subseteq N^+(u) \cup N^+(v)$. If $D = \{u, v\}$, then by our assumption for every vertex w in $V \setminus D$ there exists a vertex in $N^+(w) \cap D$. Therefore D is a dominating set of S. Since $\Delta^+(S) \leq n - 2$, $\gamma(S) > 1$ and hence $\gamma(S) = 2$. □

Using the above two characterisations we now characterise the signed graphs with domination number equal to 3.

Theorem 17 *For any signed graph S of order n, $\gamma(S) = 3$ if and only if*

(i) $\Delta^+(S) \leq n-3$,

(ii) for no pair of vertices u and v, $V \setminus \{u,v\} \subseteq N^+(u) \cup N^+(v)$ and

(iii) there exist three vertices u, v and w such that $V \setminus \{u,v,w\} \subseteq N^+(u) \cup N^+(v) \cup N^+(w)$.

Proof. To prove the necessary part suppose that $\gamma(S) = 3$. By Theorem 15 we get $\Delta^+(S) < n-1$. Now assume that $\Delta^+(S) = n-2$ and let u be a vertex with $d^+(u) = n-2$. Then the set $\{u,v\}$ is a dominating set of S, where $v \notin N^+(u)$. This contradicts the fact that $\gamma(S) = 3$. Therefore, $\Delta^+(S) \leq n-3$. If there exists a pair of vertices u and v such that $V\setminus\{u,v\} \subseteq N^+(u)\cup N^+(v)$, then $\{u,v\}$ is a dominating set of S contradicting $\gamma(S) = 3$. Now, let $D = \{u,v,w\}$ be a $\gamma(S)$-set. Then by the definition of a dominating set we find that $V\setminus\{u,v,w\} \subseteq N^+(u) \cup N^+(v) \cup N^+(w)$.

Conversely, assume that S is a signed graph satisfing the conditions (i), (ii) and (iii). Then by our assumption and Theorem 16, $\gamma(S) > 2$. Now observe that by the condition (iii) and definition of a dominating set, $\{u,v,w\}$ is a dominating set of S and hence $\gamma(S) = 3$. □

As observed earlier the signed graphs with domination number equal to n are all-negative. Now we present a necessary and sufficient conditions for a signed graph S to have $\gamma(S) = n-1$.

Theorem 18 *Let S be a signed graph of order n. Then $\gamma(S) = n-1$ if and only if S has exactly one positive edge.*

Proof. Assume that S has exactly one positive edge. Then by Remark 8, $\gamma(S) = n-1$.

Conversly, suppose that S is a signed graph with $\gamma(S) = n-1$. Let D be a $\gamma(S)$-set and $V \setminus D = \{v\}$. Then there exists a vertex u in D such that $vu \in E^+(S)$. We prove that v is incident with exactly one positive edge. On the contrary let there be another edge $vw \in E^+(S)$. Then the set $D\setminus\{u,w\}\cup\{v\}$ is a dominating set with cardinality less than that of D, which is a contradiction. Hence there exists no positive edge between the sets D and $V \setminus D$ except the edge uv. Now we claim that there is no positive edge between any two vertices x, y in D where $y \neq u$. Suppose that $xy \in E^+(S)$, then the set $D \setminus \{y\}$ is a dominating set having cardinality less than that of D, which is a contradiction. Therefore there are no positive edges between the vertices in D. Hence S has only one positive edge which is the edge uv. □

Now we characterise the signed graphs having domination number equal to $n-2$.

Theorem 19 *For any signed graph* S, $\gamma(S) = n-2$ *if and only if the subgraph induced by* $E^+(S)$ *belongs to* $\{2K_2, P_3, C_3, P_4, C_4\}$.

Proof.

To prove the sufficiency, suppose that the graph induced by the positive edges belongs to the set $\{2K_2, P_3, C_3, P_4, C_4\}$. Then by using Remark 8 $\gamma(S) = \gamma(S^+) = n-2$.

Conversely, suppose that S is a signed graph of order n and $\gamma(S) = n-2$. Let D be a $\gamma(S)$-set and $V \setminus D = \{p, q\}$. First we claim that p and q cannot be dominated by more than one vertex in D. If possible assume that the vertex p is dominated by two vertices u and v belonging to D. Then the set $D \setminus \{u, v\} \cup \{p\}$ is a dominating set with cardinality less than that of D, which is a contradiction. Similarly, we can prove that q cannot be dominated by more than one vertex in D. This shows that there are only two positive edges between the sets D and $V \setminus D$. Further, p and q are dominated by the same vertex or by two distinct vertices.

Case 1: The vertices p and q are dominated by same vertex r in D. In this case, the edges $pr, qr \in E^+(S)$. We claim that there is no positive edge between the vertices in D. On the contrary, suppose that there exists a positive edge between the vertices u and v in D, where $v \neq r$. Then observe that the set $D \setminus \{v\}$ is a dominating set having lesser cardinality than D, which is a contradiction. Now observe that there is no positive edge between the sets D and $V \setminus D$ other than pr and qr and between the vertices in D. Therefore the graph induced by the positive edges is either a P_3 or a C_3.

Case 2: The vertices p and q are dominated by two distinct vertices r and t belonging to D, respectively. Then the edges pr and qt belongs to $E^+(S)$. Using similar arguments as in ***Case 1*** we conclude that there are no positive edges between any two vertices u and v of D, where $v \neq r, t$. Thus the graph induced by the positive edges is either $2K_2$, P_4 or C_4.

Therefore the graph induced by $E^+(S)$ belongs to $\{2K_2, P_3, C_3, P_4, C_4\}$. □

References

[1] B. D. Acharya, Minus domination in signed graphs, *J. comb. inf. syst. sci.*, **37** (2009), 333–358. ⇒2

[2] B. D. Acharya, Domination and absorbance in signed graphs and digraphs, *J. Comb. Math. Comb. Comput.*, **84** (2013), 5–20. ⇒2

[3] X. Baogen, E.J. Cockayne, T. W. Haynes and S. T. Hedetniemi and Z. Shangchao, Extremal graphs for inequalities involving domination parameters, *Discrete Math.*, **216** (2000), 1–10. ⇒6
[4] C. Berge, *The Theory of Graphs and Its Applications*, Methuen, London, 1962. ⇒2
[5] G. Chartrand, H. Gavlas, F. Harary and M. Schultz, On signed degrees in signed graphs, *Czechoslov. Math. J.*, **44** (1994), 677–690. ⇒2
[6] G. Chartrand, T. W. Haynes, M. A. Henning, and P. Zhang, *From domination to coloring: Stephen hedetniemi's graph theory and beyond*, Springer International Publishing, Springer Nature Switzerland AG, 2019. ⇒2
[7] F. Harary, On the notion of balance of a signed graph, *Michigan Math. J.*, **2** (1953) 143–146. ⇒2
[8] T. W. Haynes, S. T. Hedetniemi, and M. A. Henning, *Topics in domination in graphs*, Springer International Publishing, Springer Nature Switzerland AG, 2020. ⇒2
[9] T. W. Haynes, S. T. Hedetniemi, and M. A. Henning, *Structures of domination in graphs*, Springer International Publishing, Springer Nature Switzerland AG, 2021. ⇒2
[10] T. W. Haynes, S. T. Hedetniemi, and P. J. Salter, *Domination in graphs advanced topics*, Routledge, New York, 1998. ⇒2
[11] T. W. Haynes, S. T. Hedetniemi, and P. J. Salter, *Fundamentals of domaintion in graphs*, Marcel Dekker, New York, 1998. ⇒2, 5
[12] M. A. Henning and A. Yeo, *Total domination in graphs*, Springer International Publishing, Springer New York Heidelberg Dordrecht London, 2013. ⇒2
[13] P. Jeyalakshmi, Domination in signed graphs, *Discrete Math. Algorithms Appl.*, **13** (2020). ⇒2, 3
[14] J. Joseph and M. Joseph, Roman domination in signed graphs, *Commun. comb. optim.*, https://doi.org/10.22049/cco.2022.27438.1264,(2022). ⇒2
[15] A. J. Mathias, V. Sangeetha, M. Acharya, Restrained domination in signed graphs, *Acta Univ. Sapientiae Math.*, **12** (2020), 155–163. ⇒2
[16] O. Ore, *Theory of graphs*, American Mathematical Society colloquium publications, American Mathematical Society, Providence RI (1962). ⇒3, 5
[17] H. B. Walikar, S. V. Motammanavar, and B. D. Acharya, Signed domination in signed graph, *J. comb. inf. syst. sci.*, **40** (2015), 107–128. ⇒2
[18] T. Zaslavsky, Signed graphs, *Discrete Appl. Math.*, **4** (1982), 47–74. ⇒2
[19] T. Zaslavsky, A mathematical bibliography of signed and gain graphs and allied areas, *Electron. J. Combin.*, **1000** (2018), 1–524. ⇒2

Received: February 20, 2023 • Revised: March 10, 2023

Acta Univ. Sapientiae Informatica **15,** 1 (2023) 10-24

DOI: 10.2478/ausi-2023-0002

SapiPin: Observations on PIN-code typing dynamics

Margit ANTAL
Sapientia Hungarian University of Transylvania, Cluj-Napoca, Romania
email: manyi@ms.sapientia.ro
ORCID:0000-0003-3596-1365

Krisztián BUZA
Jozef Stefan Institute
Ljubljana, Slovenia
email: chrisbuza@yahoo.com
ORCID:0000-0002-7111-6452

Abstract. In this paper, we report on PIN-code typing behaviour on touchscreen devices of 112 subjects. Detailed statistical analysis revealed that the major difference between subjects is in inter-key latency. Keypress duration variations are insignificant compared to inter-key latency variations. Subjects were grouped into meaningful clusters using clustering. The resulting clusters were of slow, medium, and fast typists. The dataset was split randomly into two equal size subsets. The first subset was used to train different synthetic data generators, while the second subset was used to evaluate an authentication attack using the generated synthetic data. The evaluation showed that slow typists were the hardest to attack. Both the dataset and the software are publicly available at https://github.com/margitantal68/sapipin_paper.

1 Introduction

While the dynamics of typing a text on a usual keyboard has been shown to be characteristic to the user, see e.g. [3, 14] and the references therein, the dynamics of typing PIN codes on numeric keyboards is somewhat understudied even though most of the wide-spread user authentication techniques are based

Key words and phrases: keystroke dynamics, PIN-code, synthetic attack, anomaly detection

on (or include the use of) PIN-codes: we may use PIN-codes to unlock our smartphone and One-Time PIN-codes (OTP) in online banking applications. Therefore, it is important to examine the typing characteristics of people when typing such short PIN-codes. PIN-codes are usually entered using a numeric keypad, which may significantly affect the typing rhythm.

Unlike previous research where all participants had to type the same PIN-code [6, 13], in our research all participants had to type a randomly generated PIN-code 20 times without error. This made it possible to observe similarities when typing different PIN-codes, and also made it possible to observe how a stable typing rhythm develops for each individual.

Our main contributions are as follows:

- We present *SapiPin*, a new dataset, which contains PIN-code typing data collected on mobile devices from 112 users.
- Exploratory data analysis through clustering revealed three meaningful typist groups.
- An attack on typing rhythm was performed using several types of synthetic data. We show the attack effectiveness for different typist groups.
- In order to assist reproduction of the results, we published our code in a public GitHub repository[1].

The rest of the paper is organized as follows. Section 2 provides a concise overview of related works. Section 3 presents the *SapiPin* dataset with the collection protocol and basic information about the subjects. Section 4 is devoted to exploratory data analysis. We begin Section 5 with a brief description of methods used to synthesize data and to detect anomalies. This is followed by the details of our evaluation protocol and our observations. Finally, we conclude in Section 6.

2 Related work

Many studies have already examined the dynamics of entering passwords. Earlier studies performed the experiments using a classic keyboard [11, 12], while the more recent ones used touch-screens of mobile phones [1, 3, 4, 2]. Moreover, Gunetti and Picardi [10] studied the usage of free text typing for continuous authentication.

[1] https://github.com/margitantal68/sapipin_paper

The dynamics of typing may be affected be various properties of the keyboard, such as its size and the placement of keys, especially whether the keyboard is alphanumeric (QWERTY) or numeric. Therefore, additionally to the research considering alphanumeric keyboards, it is important to study keystroke dynamics on numerical keyboards too.

Out of the works related to numeric keyboards, we point of the experiments of Clarke et al. [6] in which 16 subjects entered 10-digit telephone numbers and 4-digit PIN-codes. 30 samples were collected from which 20 were used for template creation and the remaining 10 for validation. They report 11.3% Equal Error Rate (EER) in an user authentication study. Better performance (8.60 % EER) was reported by Maxion and Killourhy [13] in an authentication experiment using 10-digit numbers. 28 subjects were involved in the experiment who donated 200 samples of the same 10-digit number in 4 consecutive sessions. Bours and Masoudian [5] investigated the usage of 6-digit one-time PIN- codes for user authentication. In an experiment conducted with 30 participants, they found that in the case of one-time PIN-codes the EER of authentication was 26%.

Some studies focused on the distribution of timings of keystroke dynamics. Dhakal et al. [8] conducted a study with 168,000 participants in order to find keystroke patterns linked to typing performance. They found huge differences in inter-key times between slow and fast typists, however, keypress times were very similar across these groups. Gonzales et al. [9] compared several distributions in order to rank them according to their similarity to timing histograms in free text keystroke dynamics. They found that log-logistic distributions are excellent choices for modelling the shape of timing histograms.

Timing distributions are very important when the task is to generate synthetic forgeries. Deian et al. [7] examined the robustness of keystroke-dynamics based biometrics against synthetic forgeries. Their bots generate both the keystroke duration and inter-key latency using Gaussian distributions. They reported high True Positive and low False Positive rates on a small dataset containing the data of 20 users. The good result is probably due to the weak attack, because the inter-key latency cannot be considered normally distributed.

None of the aforementioned works examined the distribution of keystroke timing in case of a numeric keyboard. This study aims to fill this gap.

Personal attribute	Amount
Age	
21–30 years	103
40–60 years	9
Gender	
Male	85
Female	27

Table 1: Personal information of the 112 subjects of the *SapiPin* dataset.

3 The *SapiPin* dataset

The *SapiPin*[2] dataset was collected in 2021 and contains 6-digit PIN typing of 112 voluntary participants. Each subject typed a randomly generated 6-digit PIN-code 20 times on his/her own mobile device. Only error-free typing was saved.

We used a web application only available on touchscreen mobile devices to collect data[3]. The application is implemented in PHP and JavaScript. Timestamps are recorded by the Date.now() JavaScript function. Subjects agree to have their data used for scientific experiments by sending the collected data to the email address provided for this purpose.

The keyboard layout of this data collector is shown in Fig. 1. Detailed personal information of the subjects is presented in Table 1.

4 Preprocessing and Data Analysis

4.1 Preprocessing

The collected dataset was preprocessed to remove corrupted items. We excluded the data of a single person, the user with ID number 96, in case of whom it happened several times that the timestamps of pressing were identical to the timestamps of release which indicates an error with recording the data. As a result, we work with the data of 111 users from now on (The database published in 2022 still contains the data of user with ID 96).

In the next step, we computed two types of features: (i) keypress duration, a.k.a. hold time, and (ii) inter-key latency, i.e., the duration between releasing

[2] https://github.com/margitantal68/sapipin

[3] source code: https://github.com/gzsolt11/PinLogger.git

Figure 1: PIN logger.

a key and pressing the next key. Therefore, in the case of a 6-digit PIN-code, the following features were computed: HT_1, HT_2, HT_3, HT_4, HT_5, HT_6 (hold times), and RP_1, RP_2, RP_3, RP_4, RP_5 (release-press times).

The next step in preprocessing was outlier detection. We found no outliers regarding keypress duration. In contrast, there were quite a few outliers for inter-key intervals. In such cases, users appeared to pause while typing. We declared a value of a feature as an outlier if it was greater than $\mu + 10\sigma$, where μ and σ are the average and the standard deviation of the feature. For the subsequent analysis, outliers were always replaced by the average of the corresponding feature of the given user. In total, 13 outliers were replaced. These outliers belonged to the following 11 distinct users: 2, 11, 20, 21, 25, 48, 53, 58, 73, 77, 93. In most cases, each user exhibited a single instance of an outlier feature within their respective samples, with the exception of users 25 and 77, who each presented two instances. Of the 13 detected cases, 70% involved the user's first sample as the source of the outlier feature.

The final dataset contains 13320 keystrokes from 111 participants (120 keystrokes per participant). Boxplots of keypress duration and inter-key latency are shown in Fig. 2.

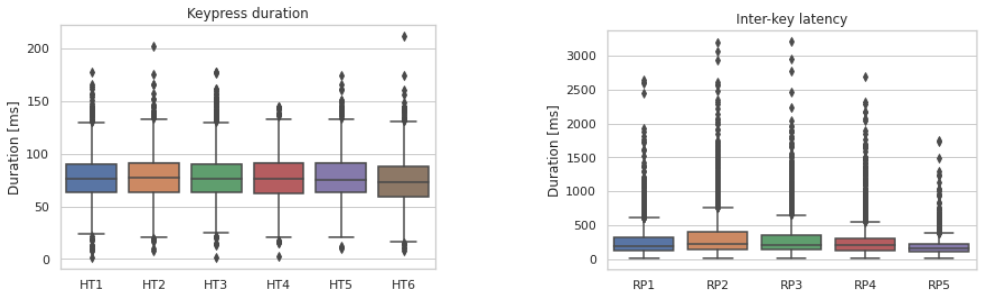


Figure 2: Keypress duration and inter-key latency boxplots

4.2 Data analysis

Outlier correction was followed by descriptive statistics. Keypress duration and inter-key intervals were analyzed separately. Fig. 3 shows the histograms of keypress duration (HT hold times) and inter-key interval (release-press times). The average keypress duration is 76.89 ms with a standard deviation of 22.17. The skewness of the distribution is 0.40, while its kurtosis is 0.64. In contrast, the average inter-key interval is 269.56 ms with standard deviation, skewness and kurtosis of 266.67, 3.78 and 21.36 respectively. It can be observed that the inter-key interval average is more than three times higher than keypress duration average. The high positive skewness of the inter-key interval distribution is inline with the observations in [8]. However, we observed an even higher skewness than in case of usual keyboards.

In a previous study [4], we observed that the average keypress duration, as well as the average inter-key interval is a discriminative feature for user authentication based on keystroke dynamics. Therefore, we computed three new features of PIN-code typing for each user: (i) $\mathsf{AVG}_{\mathsf{HT}}$, the average of keypress duration, (ii) $\mathsf{AVG}_{\mathsf{RP}}$, the average of inter-key interval, and (iii) $\mathsf{TOTAL}_{\mathsf{TIME}}$, the duration of typing the entire PIN code. In the next step, correlations were computed between these features: while the correlations between $\mathsf{AVG}_{\mathsf{HT}}$ and the two other features are close to zero ($\mathsf{corr}(\mathsf{AVG}_{\mathsf{HT}}, \mathsf{AVG}_{\mathsf{RP}}) = -0.04$, $\mathsf{corr}(\mathsf{AVG}_{\mathsf{HT}}, \mathsf{TOTAL}_{\mathsf{TIME}}) = 0.08$), $\mathsf{AVG}_{\mathsf{RP}}$ strongly correlates with $\mathsf{TOTAL}_{\mathsf{TIME}}$ ($\mathsf{corr}(\mathsf{AVG}_{\mathsf{RP}}, \mathsf{TOTAL}_{\mathsf{TIME}}) = 0.99$). We can state that the typing speed clearly depends only on the inter-key latency. Fig. 4 shows the uncorrelated nature of the $\mathsf{AVG}_{\mathsf{HT}}$ and $\mathsf{AVG}_{\mathsf{RP}}$ features.

We investigated how the typing rhythm of PIN code of each user evolves over time. To this end, we computed the ratio of $\mathsf{AVG}_{\mathsf{RP}}$ to $\mathsf{AVG}_{\mathsf{HT}}$ for each PIN

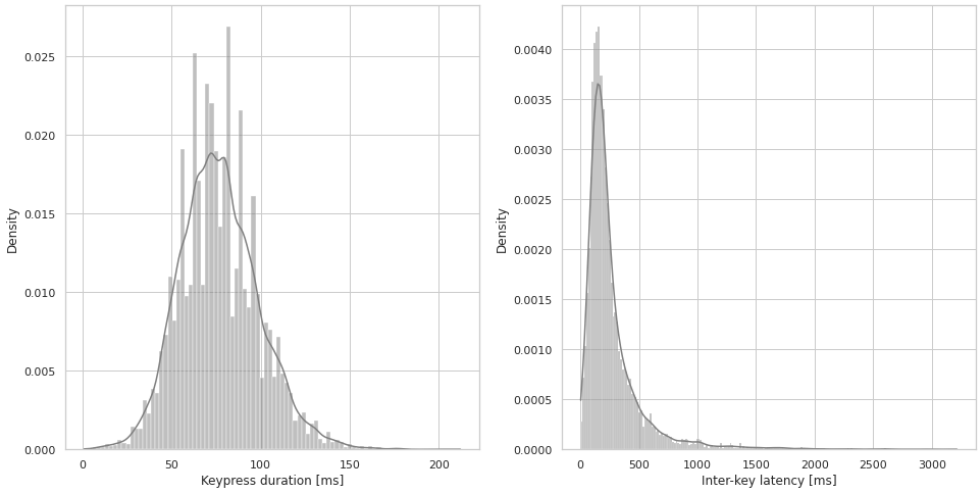


Figure 3: Keypress duration and inter-key latency histograms

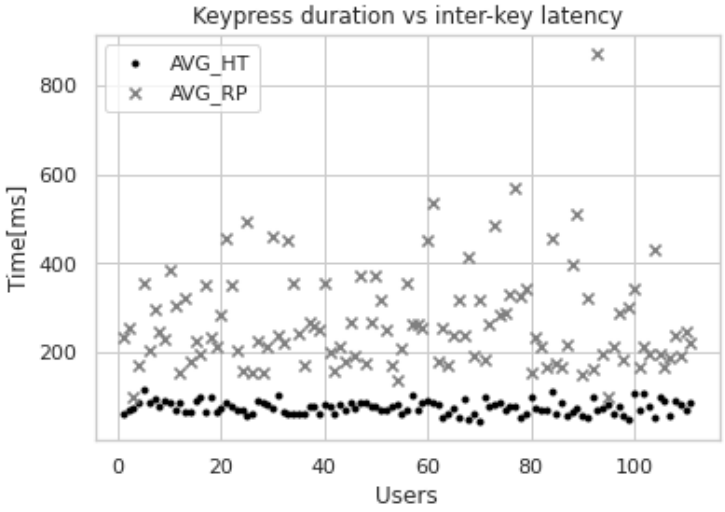


Figure 4: Users AVG_{HT} and AVG_{RP}

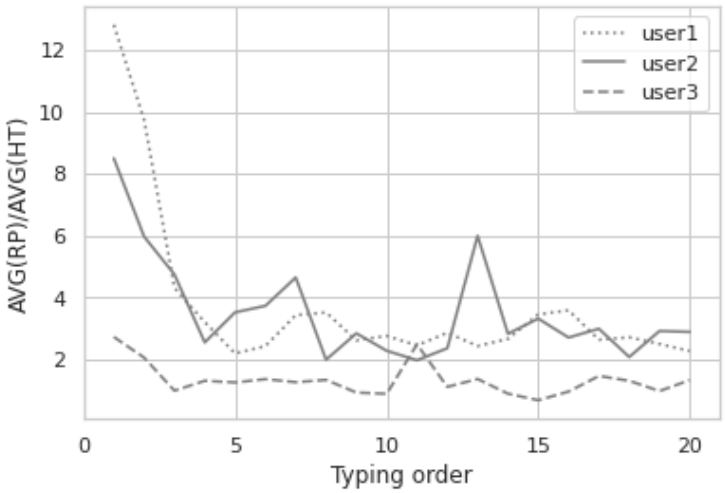
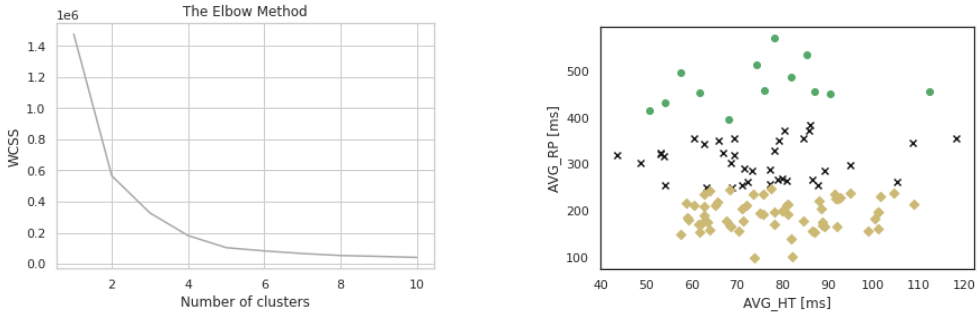


Figure 5: Rhythm of typing stabilises over time

typing. Our analysis revealed that the first two samples typically exhibited significant differences from subsequent samples. However, once these initial samples were completed, a more consistent rhythm emerged with relatively smaller variations, as depicted in Figure 5.

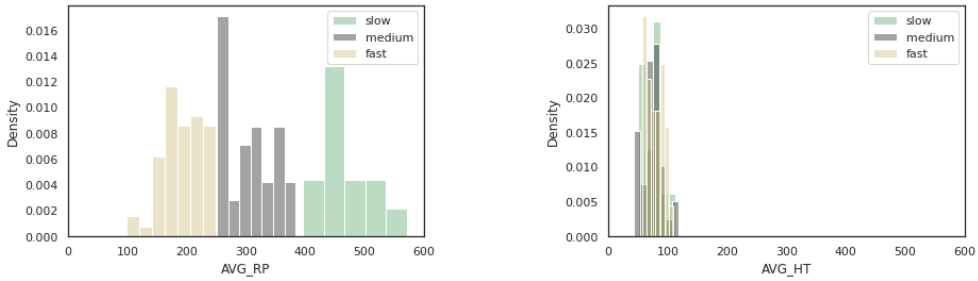
4.3 Clustering

The objective of clustering is to find meaningful groups within a dataset. Clustering belongs to unsupervised learning: unlabelled instances are grouped (or the labels of instances are not used when the groups are determined). In our case, we expect to see clusters of users having similar typing characteristics. The original dataset was converted to a reduced dataset, where each user is represented by a single aggregated instance, which contains only two features AVG_{HT} and AVG_{RP} corresponding to the average hold time and average inter-key latency of the user. K-means was used as the clustering algorithm. In order to find the most suitable number of clusters, we used the elbow method. Fig. 6a shows the result of the elbow method, which indicates that this dataset contains four meaningful groups. We have one group containing only one user (userid=93), who has extremely large inter-key latency (this can be seen in Fig.4 too). We excluded this user from further analysis. We denote the three other groups as slow (13 users, green dots), medium (37 users, black x marker) and fast (60 users, yellow diamond marker) typists. It can be seen in Fig. 6b



(a) Elbow method (b) User groups

Figure 6: K-means clustering

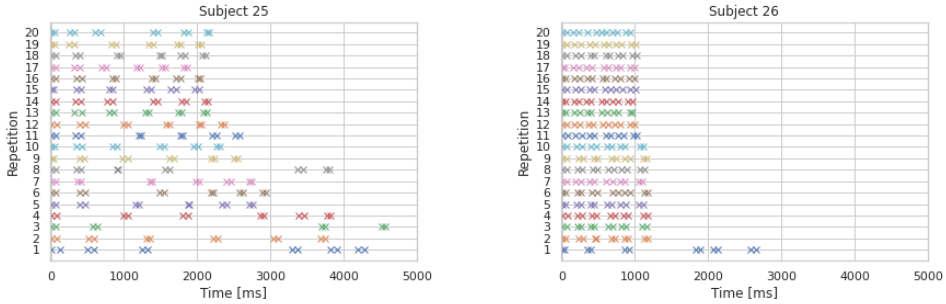


(a) Clusters' $\mathsf{AVG}_{\mathsf{RP}}$ histogram. (b) Clusters' $\mathsf{AVG}_{\mathsf{HT}}$ histogram.

Figure 7: K-means clustering

that these three groups are separated by the $\mathsf{AVG}_{\mathsf{RP}}$ feature. The typing patterns of a slow typist and a fast typist are illustrated in Figure 8.

During the preprocessing phase of the data analysis, a total of 13 outliers were identified and attributed to 11 unique users. Specifically, one of the users (user ID=93) exhibited a significantly slow typing speed. The remaining 10 users were distributed across three typing categories, with four users categorized as slow typists (IDs 21, 25, 73, and 74), four users classified as medium typists (IDs 2, 11, 20, and 58), and two users classified as fast typists (IDs 48 and 53). These findings indicate that outliers were present across all levels of typing proficiency. As noted in the data analysis chapter, it was observed that outliers were more frequently detected within the initial typing pattern of each user.



(a) Slow typist. (b) Fast typist.

Figure 8: Typing patterns

5 Attack framework

5.1 Synthetic data generation

The SDV package was used to generate synthetic data[4]. We used three types of models: Gaussian Copula, Conditionally Time Generative Adversarial Network (CTGAN), and Time Variational Autoencoder (TVAE). The CTGAN and TVAE models were proposed by Lei Xu et al. in a paper presented in 2019 at the NeurIPS conference [15].

A copula in mathematical terms is a distribution over $[0, 1]^d$ unit cube, created by using the probability integral transform from a multivariate normal distribution over R^d. Essentially, a copula is a mathematical function that describes the joint distribution of multiple random variables by examining the dependence between their marginal distributions.

CTGAN is a method that uses GANs to model the distribution of tabular data and generate samples from it. It overcomes the issues of non-Gaussian and multimodal distributions with mode-specific normalization. CTGAN employs a conditional generator and training-by-sampling. High-quality models are trained with fully-connected networks and several recent techniques [15].

The third synthetic data generation method is a special variational autoencoder, named TVAE, as this is an adaptation of variational autoencoders to tabular data.

Synthetic data generators must be trained and the data used for their training should not be used in further evaluations. We solved this issue by dividing the *SapiPin* dataset into two subsets: we used the data of 55 randomly selected

[4] https://sdv.dev/

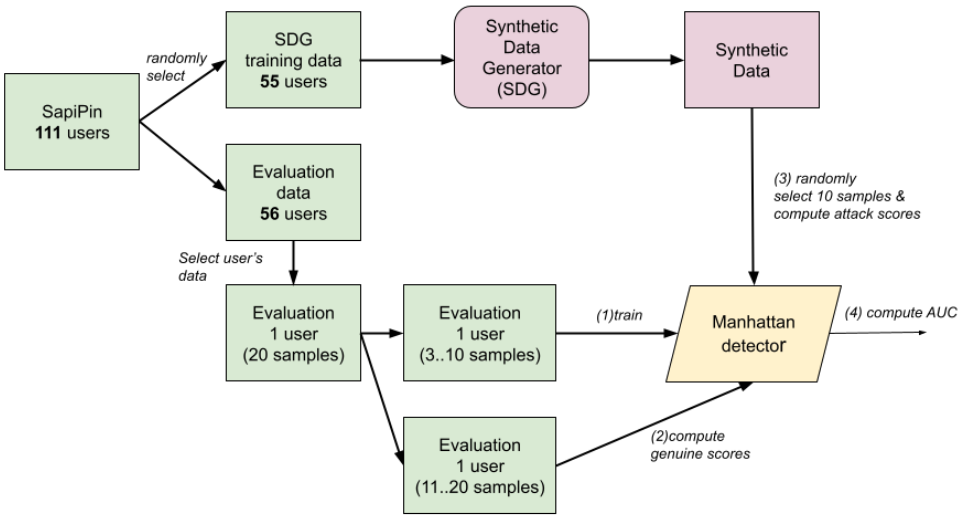


Figure 9: Evaluation of PIN-code attack framework

users to train the synthetic data generator, and we evaluated our attack framework on the data of the remaining 56 users (see the inital splitting of the data in the top-left of Figure 9).

5.2 Anomaly detector

We evaluate synthetic data quality via anomaly detector. High-quality synthetic data is difficult to detect. There are many different types of anomaly detectors. In the case of keystroke dynamics-based user authentication, one of the well-established detectors is the scaled Manhattan [12].

We trained a separate model for each user of the dataset using only real user data, and then evaluated it on both real and synthetic data. Based on the real and synthetic score values obtained from the evaluation results, we calculated the area under the curve (AUC). Average and standard deviation of these AUCs are reported for the 56 users in the evaluation dataset (see 4th step in Figure 9).

High quality synthetic data is difficult to detect by the anomaly detector, hence a lower AUC will be obtained in the case of better synthetic data.

5.3 Evaluation protocol and results

Each user in the dataset has 20 typing examples. As we observed that the first two examples are significantly different from the remaining ones, see Fig. 5, we excluded these two from anomaly detector training. Therefore, the first two examples were dropped, and only 8 genuine examples were used for training the anomaly detector. The remaining 10 examples were used for testing the anomaly detector, obtaining 10 positive (genuine) scores. We randomly selected 10 examples from the synthetic data as fraudulent examples, and presented them as an input to the anomaly detector. This resulted in 10 negative or attack scores. Based on the negative and positive score values, we calculated the AUC value for each user. The evaluation was repeated 10 times, each time using other randomly selected synthetic data for attack. Our synthetic data was generated based on training data which contains typing of different PIN-codes, therefore its samples are general typing rhythms of 6-digit PIN-codes. In our attack model we assume that the attacker knows the subject's PIN-code, but does not know its typing rhythm. Therefore, it uses a random sample from synthetic typing rhythms.

Table 2 reports the mean AUC and its standard deviation for each type of synthetic data. These results suggest that, compared with the synthetic data generated by CTGAN, both Gaussian Copula and TVAE generated data were highly similar to the genuine data, because in these cases it was more difficult to detect for the anomaly detector whether the data is real or synthetic, as indicated by the lower AUC value of Gaussian Copula and TVAE compared with that of CTGAN.

Synthetic data generation method	AUC (std)
Gaussian Copula	0.90 (0.10)
CTGAN	0.98 (0.03)
TVAE	0.89 (0.11)

Table 2: Synthetic attacks on the *SapiPin* dataset.

The present study aimed to investigate the potential relationship between typing speed and AUC values. To this end, we analyzed the average typing time per participant, derived from eight training samples, in conjunction with the AUC values obtained from TVAE synthetic data (see Fig. 10). Each data point in the figure corresponds to a specific participant. Our findings reveal that

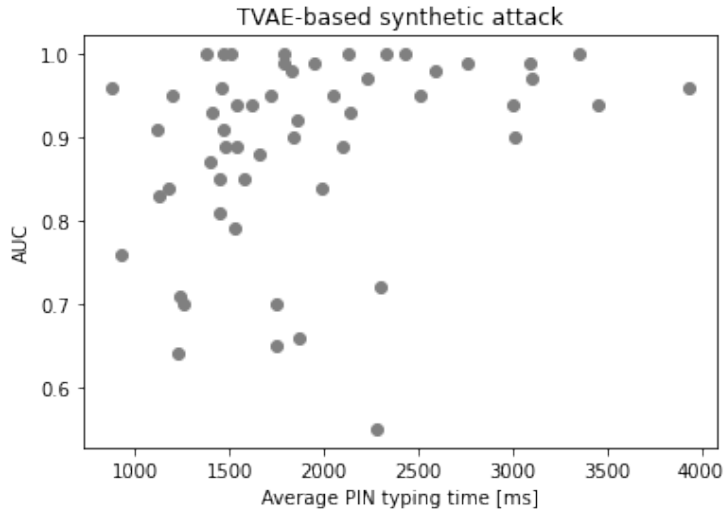


Figure 10: TVAE-based synthetic attack: users average typing time vs. AUC.

the AUC values attained by the slow typists were found to be comparatively elevated and exhibit a low degree of standard deviation. Although the majority of fast typists exhibit relatively high AUC values, there are several individuals in this group who demonstrate lower AUC values compared to the overall average.

5.4 Evaluation on typist groups

This section presents the outcomes of our investigation regarding three distinct typist groups, namely slow, medium, and fast typists. To this end, we applied the protocol described in the preceding section, with the evaluations carried out thrice, once for each typist group. Our findings, summarized in Table 3, highlight the highest degree of detection success among the slow typists, implying that they represent the most challenging target for potential attackers.

6 Conclusions

In this paper we presented *SapiPin*, a new dataset of 6-digit PIN-code typing on touchscreens. Data analysis revealed that the data can be divided into meaningful groups belonging to slow, medium, and fast typists. The major

Synthetic data generation method	AUC (std) Slow	Medium	Fast
Gaussian Copula	0.93 (0.08)	0.89 (0.12)	0.90 (0.09)
CTGAN	0.98 (0.03)	0.98 (0.04)	0.98 (0.03)
TVAE	0.95 (0.05)	0.89 (0.12)	0.88 (0.10)

Table 3: Synthetic attacks on the typist groups.

difference between these groups is the inter-key latency. This observation confirms the results of a previous study [8] in which the same observations were reported for free text typing. Moreover, the reported timing distributions are similar to ours, even though only a 10-digit software keyboard was used in our study.

The dataset was split randomly into equally sized subsets. Half of the data was used to train different synthetic data generators. The best quality synthetic data (the most similar to the original one) was generated by the TVAE method. Using the synthetic data, we performed synthetic attacks on each user from the second half of the *SapiPin* dataset. Results show that slow typists are the hardest to attack. At the same time, in this group we can see the greatest dispersion among the typing samples of a user.

We acknowledge limitations regarding the generalizability of the results, as young, European participants (university students) are over-represented in the data.

Acknowledgements

We would like to thank Zsolt Gergely for creating the data collection application and for his assistance with data collection.

References

[1] J. Angulo, E. Wästlund. Exploring touch-screen biometrics for user identification on smart phones. In J. Camenisch, B. Crispo, S. Fischer-Hübner, R. Leenes, G. Russello, editors, *Privacy and Identity Management for Life*, pages 130–143, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ⇒11

[2] M. Antal, L. Nemes. The mobikey keystroke dynamics password database: Benchmark results. In *Software Engineering Perspectives and Application in Intelligent Systems*, pages 35–46, Cham, 2016. Springer International Publishing. ⇒11

[3] M. Antal, L. Z. Szabo, I. Laszlo. Keystroke dynamics on android platform. *Procedia Technology*, **19** (2015) 820–826. 8th Int. Conf. Interdisciplinarity in Engineering, INTER-ENG 2014, 9–10 October 2014, Târgu Mureș, Romania. ⇒10, 11

[4] M. Antal, L. Z. Szabó. An evaluation of one-class and two-class classification algorithms for keystroke dynamics authentication on mobile devices. In *20th Int. Conf. on Control Systems and Computer Sci.*, pages 343–350, 2015. ⇒11, 15

[5] P. Bours, E. Masoudian. Applying keystroke dynamics on one-time pin codes. In *2nd Int. Workshop on Biometrics and Forensics*, pages 1–6, 2014. ⇒12

[6] N. Clarke, S. Furnell, B. Lines, P. Reynolds. Keystroke dynamics on a mobile handset: a feasibility study. *Information Management & Computer Security*, **11,** 4 (2003) 161–166. ⇒11, 12

[7] S. Deian, S. Xiaokui, D. Danfeng. Robustness of keystroke-dynamics based biometrics against synthetic forgeries. *Computers & Security*, **31,** 1 (2012) 109–121. ⇒12

[8] V. Dhakal, A. M. Feit, P. O. Kristensson, A. Oulasvirta. Observations on typing from 136 million keystrokes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–12, New York, NY, USA, 2018. Association for Computing Machinery. ⇒12, 15, 23

[9] N. González, E. P. Calot, J. S. Ierache, W. Hasperué. On the shape of timings distributions in free-text keystroke dynamics profiles. *Heliyon*, **7,** 11 (2021) e08413. ⇒12

[10] D. Gunetti, C. Picardi. Keystroke analysis of free text. *ACM Trans. Inf. Syst. Secur.*, **8,** 3 (2005) 312–347. ⇒11

[11] R. Joyce, G. Gupta. Identity authentication based on keystroke latencies. *Commun. ACM*, **33,** 2 (1990) 168–176. ⇒11

[12] K. S. Killourhy, R. A. Maxion. Comparing anomaly-detection algorithms for keystroke dynamics. In *2009 IEEE/IFIP Int. Conf. on Dependable Systems Networks*, pages 125–134, 2009. ⇒11, 20

[13] R. A. Maxion, K. S. Killourhy. Keystroke biometrics with number-pad input. In *2010 IEEE/IFIP Int. Conf. on Dependable Systems & Networks (DSN)*, pages 201–210, 2010. ⇒11, 12

[14] D. Neubrandt, K. Buza. Pro‚jection-based person identification. In *Proceedings of the 10th Int. Conf. on Computer Recognition Systems CORES 2017*, pages 221–228. Springer, 2018. ⇒10

[15] L. Xu, M. Skoularidou, A. Cuesta-Infante, K. Veeramachaneni. Modeling tabular data using conditional gan. In *Proceedings of the 33rd Int. Conf. on Neural Information Processing Systems*, pages 7335–7345, 2019. ⇒19

Received: March 13, 2023 • Revised: April 11, 2023

Acta Univ. Sapientiae Informatica **15,** 1 (2023) 25–37

DOI: 10.2478/ausi-2023-0003

Connected certified domination edge critical and stable graphs

Azham ILYASS LONE
Chandigarh University
Mohali, Punjab, India
email: aazamlone22@gmail.com

Vishwajeet GOSWAMI
Chandigarh University
Mohali, Punjab, India
email:
vishwajeetgoswami.math@gmail.com

Abstract. In an isolate-free graph $\mathcal{Z} = (V_{\mathcal{Z}}, E_{\mathcal{Z}})$, a set $\mathcal{C}$ of vertices is termed as a connected certified dominating set of $\mathcal{Z}$ if, $|N_{\mathcal{Z}}(u) \cap (V_{\mathcal{Z}} \backslash \mathcal{C})| = 0$ or $|N_{\mathcal{Z}}(u) \cap (V_{\mathcal{Z}} \backslash \mathcal{C})| \geq 2$ $\forall u \in \mathcal{C}$, and the subgraph $\mathcal{Z}[\mathcal{C}]$ induced by $\mathcal{C}$ is connected. The cardinality of the minimal connected certified dominating set of graph $\mathcal{Z}$ is called the connected certified domination number of $\mathcal{Z}$ denoted by $\gamma^{c}_{cer}(\mathcal{Z})$. In graph $\mathcal{Z}$, if the deletion of any arbitrary edge changes the connected certified domination number, then we call it a connected certified domination edge critical. If the deletion of any random edge does not affect the connected certified domination number, then we refer to it as a connected certified domination edge stable graph. In this paper, we investigate those graphs which are connected certified domination edge critical and stable upon edge removal. We then study some properties of connected certified domination edge critical and stable graphs.

1 Introduction

Detlaff et al. [8] introduced certified domination in 2020, and it is now a well-studied domination-related parameter in the domination theory of graphs

Key words and phrases: connected certified domination, connected certified domination edge critical, connected certified domination edge stable.

(see, for example, [9, 15, 16, 14] for recent literature on this topic). A set $\mathcal{C} \subseteq V_{\mathcal{Z}}$ of a graph $\mathcal{Z} = (V_{\mathcal{Z}}, E_{\mathcal{Z}})$ is called certified dominating set (CFDS) if $|N_{\mathcal{Z}}(u) \cap (V_{\mathcal{Z}} \backslash \mathcal{C})| = 0$ or $|N_{\mathcal{Z}}(u) \cap (V_{\mathcal{Z}} \backslash \mathcal{C})| \geq 2$ $\forall u \in \mathcal{C}$. The cardinality of the minimal CFDS is the certified domination number (CFDN) of the graph $\mathcal{Z}$, represented by $\gamma_{cer}(\mathcal{Z})$ [8]. A $\gamma_{cer}-$ set $\mathcal{C}$ is said to be a connected certified dominating set (CCDS), if the induced subgraph $\mathcal{Z}[\mathcal{C}]$ is connected and $|N_{\mathcal{Z}}(u) \cap (V_{\mathcal{Z}} \backslash \mathcal{C})| = 0$ or $|N_{\mathcal{Z}}(u) \cap (V_{\mathcal{Z}} \backslash \mathcal{C})| \geq 2$ $\forall u \in \mathcal{C}$. The connected certified domination number (CCDN) of the graph $\mathcal{Z}$ is the cardinality of the smallest CCDS and is represented by $\gamma^c_{cer}(\mathcal{Z})$. An element $u \in V_{\mathcal{Z}}$ is a $\gamma^c_{cer}-$ good vertex if u is in some $\gamma^c_{cer}-$ set of the graph $\mathcal{Z}$, and set of all $\gamma^c_{cer}-$ good vertices of the graph $\mathcal{Z}$ will be represented by $T^c_{cer}(\mathcal{Z})$.

Criticality and stability are important considerations for a lot of graph parameters. It is generally essential to understand how a graphical property behaves when the graph is altered when it is relevant in an application. Much has been written on graphs where the deletion (addition) of an edge (vertex) affects a parameter (such as domination number or chromatic number). The $\gamma-$critical graphs when one edge is eliminated were examined by Walikar and Acharya [17] and in contrast, Dutton and Brigham first studied $\gamma-$ stable graphs [10]. These problems were then used to investigate critical and stable graphs with respect to different domination variations such as, "Roman Domination", "Total Domination", "Connected Domination", etc. γ_c-critical graphs were first studied by [5] in 2004, while γ_c-stable graphs were first studied by [7] in 2015. In 2020, Detlaff et al. [8] studied the influence of edge addition and deletion on the CFDN of graphs.

The criticality and stability of graph upon edge addition and deletion have been studied for various domination-related parameters, for example, [6, 2, 4, 12]. In this research, we investigate those graphs where the CCDN increases when an edge is deleted. We also study those graphs where CCDN remains unchanged on the deletion of an edge. To analyse stable or critical graphs when one edge is eliminated, we state that $\gamma^c_{cer}(\mathcal{Z}) = \infty$ if a graph $\mathcal{Z}$ contains an isolated vertex. Consequently, $\gamma^c_{cer}(\mathcal{Z} - e) = \infty$ if we delete an edge $e \in E_{\mathcal{Z}}$ that is incident with a leaf vertex in $\mathcal{Z}$. In addition, $\gamma^c_{cer}(\mathcal{Z} - e) = \infty$ if edge $e \in E_{\mathcal{Z}}$ divides the graph $\mathcal{Z} - e$ into two components.

We state that a graph $\mathcal{Z}$ is connected certified domination edge ($ccde$) stable or $[\gamma^c_{cer}]^{e^-}$-stable, if $\gamma^c_{cer}(\mathcal{Z} - e) = \gamma^c_{cer}(\mathcal{Z})$ $\forall e \in E_{\mathcal{Z}}$. If $\gamma^c_{cer}(\mathcal{Z}) = k$, and $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-stable, then $\mathcal{Z}$ is $[k^c_{cer}]^{e^-}$-stable. A graph $\mathcal{Z}$ is $ccde$ critical or $[\gamma^c_{cer}]^{e^-}$-critical , if $\gamma^c_{cer}(\mathcal{Z} - e) \neq \gamma^c_{cer}(\mathcal{Z})$ $\forall e \in E_{\mathcal{Z}}$. We note that eliminating an edge of a graph $\mathcal{Z}$ cannot decrease the CCDN of the graph $\mathcal{Z}$. Hence if $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-

critical, then $\gamma^c_{cer}(\mathcal{Z}-e) > \gamma^c_{cer}(\mathcal{Z})$ for every edge $e \in E_{\mathcal{Z}}$. If $\gamma^c_{cer}(\mathcal{Z}) = k$, and $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical, we say that $\mathcal{Z}$ is $[k^c_{cer}]^{e^-}$-critical.
An edge $e \in E_{\mathcal{Z}}$ is a critical edge of $\mathcal{Z}$ if $\gamma^c_{cer}(\mathcal{Z}-e) > \gamma^c_{cer}(\mathcal{Z})$, whereas an edge $e \in E_{\mathcal{Z}}$ is a stable edge of $\mathcal{Z}$ if $\gamma^c_{cer}(\mathcal{Z}-e) = \gamma^c_{cer}(\mathcal{Z})$. Thus, if $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical graph, then every edge of the graph $\mathcal{Z}$ is a critical edge, while every edge in a $[\gamma^c_{cer}]^{e^-}$-stable graph is a stable edge.

1.1 Definitions and notations

We refer to [13] and [18] for general graph-theoretic definitions and notations. Throughout this paper, by a graph $\mathcal{Z}$ we mean a connected, undirected, and unweighted simple graph (i.e., graph without loops or multiple edges). A graph $\mathcal{Z} = (V_{\mathcal{Z}}, E_{\mathcal{Z}})$ with no isolated vertex is an isolate-free graph. The order of $\mathcal{Z}$ is denoted by $n(\mathcal{Z}) = |V_{\mathcal{Z}}|$ and size of $\mathcal{Z}$ by $m(\mathcal{Z}) = |E_{\mathcal{Z}}|$. For any vertex $u \in \mathcal{Z}$, $d_{\mathcal{Z}}(u)$ will denote the degree of u in $\mathcal{Z}$. The neighborhood of u, represented by $N_{\mathcal{Z}}(u)$, is the set of all nodes adjacent to u, and the degree of u in $\mathcal{Z}$ is $|N_{\mathcal{Z}}(u)|$.

Vertex $u \in \mathcal{Z}$ is called an isolated vertex if $d_{\mathcal{Z}}(u) = 0$ and is called a pendant or leaf if $d_{\mathcal{Z}}(u) = 1$. $\delta(\mathcal{Z}), (\Delta(\mathcal{Z}))$ denotes the minimal (maximal) degree among the vertices of $\mathcal{Z}$. The diameter of a graph is the largest distance between two vertices and the maximum distance between $x \in V_{\mathcal{Z}}$, and all other vertices is the eccentricity of the vertex.A universal vertex of a graph $\mathcal{Z}$ is a vertex of degree $|V_{\mathcal{Z}}| - 1$. A leaf is a degree one vertex whose only neighbor is referred to as a support vertex. A support vertex is strong if it has at least two leaves as neighbors; otherwise, it is considered weak. We will use $L_{\mathcal{Z}}$ and $S_1(\mathcal{Z})(S_2(\mathcal{Z})$, respectively) to represent the set of leaves and weak supports (strong supports, respectively) of graph $\mathcal{Z}$. For a connected graph $\mathcal{Z}$, a vertex $u \in V_{\mathcal{Z}}$ is called a cut vertex if $\mathcal{Z} - u$ is not connected. The number of cut vertices of $\mathcal{Z}$ is denoted by $\zeta(\mathcal{Z})$.
The set $N_{\mathcal{Z}}(u) \cup \{u\} = N_{\mathcal{Z}}[u]$ is a closed neighborhood of u. More specifically, the neighborhood (closed, respectively) of a subset $A \subseteq V_{\mathcal{Z}}$ of vertices, represented by $N_{\mathcal{Z}}(A)$ (resp. $N_{\mathcal{Z}}[A]$), is defined to be the set $\bigcup_{u \in A} N_{\mathcal{Z}}(u)$ (resp. $N_{\mathcal{Z}}(A) \cup A$). Let $\mathcal{C} \subseteq V_{\mathcal{Z}}$ and $u \in \mathcal{C}$. The $\mathcal{C}-$private neighborhood of u denoted by $pn(u, \mathcal{C})$, and is defined by $pn(u, \mathcal{C}) = N_{\mathcal{Z}}[u] - N_{\mathcal{Z}}[\mathcal{C} - u]$. Thus if $w \in pn(u, \mathcal{C})$, then $N_{\mathcal{Z}}(w) \cap \mathcal{C} = \{u\}$. We refer to a vertex $w \in pn(u, \mathcal{C})$ as a $\mathcal{C}-$private neighborhood of u. We construct the set $epn(u, \mathcal{C}) = pn(u, \mathcal{C}) \cap (V - \mathcal{C})$ and designate a vertex $y \in epn(u, \mathcal{C})$ an external $\mathcal{C}-$private neighbor of u. If the context makes the graph $\mathcal{Z}$ clear, we simply write $N(u)$ and $N[u]$ instead

of $N_{\mathcal{Z}}(u)$ and $N_{\mathcal{Z}}[u]$, respectively.
The star graph $S_{(1,r)}$ of order $n = r+1$, "is a tree on n vertices with one vertex having degree $|V_{(S_{(1,r)})}| - 1$ and the other $n-1$ vertices having vertex degree 1". Double star graph $S_{(q,r)}$ is a graph obtained by joining center vertex of two star graphs $S_{(1,q)}$ and $S_{(1,r)}$ with an edge. A tree in which every vertex is on a central spine or is just one edge away from the spine is known as a caterpillar graph, caterpillar tree, or simply a caterpillar (in other words, deleting its endpoints results in a path graph). "The corona product of two graphs, $\mathcal{H}_1$ and $\mathcal{H}_2$, is defined as the graph attained by taking one replica of $\mathcal{H}_1$ and $|V_{\mathcal{H}_1}|$ replicas of $\mathcal{H}_2$ and linking the j^{th} vertex of $\mathcal{H}_1$ to every vertex in the j^{th} replica of $\mathcal{H}_2$"[1, 3, 11].

2 $[\gamma^c_{cer}]^{e^-}$-critical graphs

We provide the characterization of "$[\gamma^c_{cer}(\mathcal{Z})]^{e^-}$-critical graphs" in this section. Before moving on to the key findings, we first define the family of trees $\mathcal{T}$ as follows.
We define $\mathcal{T}$ as a family of trees in which each vertex is a leaf or of degree at least 3. A tree $\mathcal{T} \in \mathcal{T}$, if $\mathcal{T}$ is a non-trivial star $S_{(1,r)}$, $r \geq 2$, or $\mathcal{T}$ is double star graph $S_{(q,r)}$, $q, r \geq 2$, or $\mathcal{T}$ is a caterpillar, or if $\mathcal{T}$ can be constructed by subdivided star $S_{(1,r)}$, $r \geq 2$ by adding zero or at least two vertices to the vertices of degree 1, or if $\mathcal{T}$ can be constructed by subdivided double star graph $S_{(q,r)}$, $q, r \geq 2$ by adding at least two vertices to the vertices of degree 1.
We begin this section with the following observation.

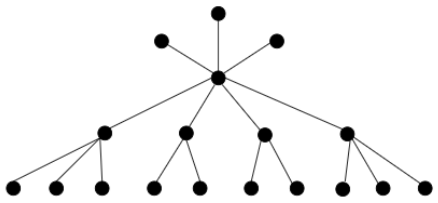


Figure 1: A graph T_{18} in the family of trees $\mathcal{T}$.

Observation 1 If $\mathcal{C}$ is the smallest γ^c_{cer}—set of a graph $\mathcal{Z}$, then for each vertex $u \in \mathcal{C}$, $|epn(u, \mathcal{C})| \geq 1$.
Note that if $\mathcal{C}$ is the smallest γ^c_{cer}—set of a connected graph $\mathcal{Z} = (V_{\mathcal{Z}}, E_{\mathcal{Z}})$ such that $\mathcal{C} = V_{\mathcal{Z}}$, then $epn(u, \mathcal{C}) = \phi$.

Proposition 1 *Let $\mathcal{Z}$ be an isolate free graph of order n with $dia(\mathcal{Z}) \leq 2$ and $\delta(\mathcal{Z}) = 1$, then $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical.*

Proof. Let $\mathcal{Z}$ be any connected graph of order n, with $dia(\mathcal{Z}) \leq 2$ and $\delta(\mathcal{Z}) = 1$, and let $\mathcal{C}$ be the γ^c_{cer}—set of the graph $\mathcal{Z}$. We will prove it in two cases.
Case 1. When $dia(\mathcal{Z}) = 2$.
If a graph $\mathcal{Z}$ has diameter two, then every pair of non-adjacent vertices has a common neighbor, and $\gamma^c_{cer}(\mathcal{Z}) = 1$, whenever $dia(\mathcal{Z}) = 2$ and $\delta(\mathcal{Z} = 1$. Let $e = uv \in \mathcal{Z}$ be such that either $u \in \mathcal{C}$ or $v \in \mathcal{C}$. Therefore, if we delete edge $e = uv$ from $\mathcal{Z}$, then the CCDN of the graph $\mathcal{Z} - e$ will change, and we know that deletion of an edge from any arbitrary graph cannot decrease its CCDN. Therefore $\gamma^c_{cer}(\mathcal{Z} - e) > \gamma^c_{cer}(\mathcal{Z})$, implying that $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical.
Case 2. When $dia(\mathcal{Z}) = 1$.
If a graph $\mathcal{Z}$ has a diameter 1, then $\mathcal{Z}$ is a path graph P_2 on two vertices u and v connected with only edge $e = uv \in \mathcal{Z}$. Now if we remove this edge e then the resultant graph will be a disconnected graph implying that $\gamma^c_{cer}(P_2 - e) = \infty$ and hence $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical.
Hence from Case 1 and case 2, we conclude that $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical. □

Proposition 2 *If $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$- critical graph and $dia(\mathcal{Z}) \leq 2$, then for every $\gamma^c_{cer}(\mathcal{Z})$—set $\mathcal{C}$ of $\mathcal{Z}$, $\mathcal{Z}[\mathcal{C}]$ is either a trivial graph or a star graph.*

The CCDN of a an isolate free graph $\mathcal{Z}$ with $dia(\mathcal{Z}) \leq 2$ in most of the cases is one and is two in the only case when $\mathcal{Z} \cong P_2$. So in cases when $\gamma^c_{cer}(\mathcal{Z}) = 1$, the subgraph induced by γ^c_{cer}—set of graph $\mathcal{Z}$ is a trivial graph, and when $\gamma^c_{cer}(\mathcal{Z}) = 2$ the subgraph induced by γ^c_{cer}—set is a star graph. Also, if a graph $\mathcal{Z}$ has $\delta(\mathcal{Z}) = 1$, then $\mathcal{Z}$ is always $[\gamma^c_{cer}]^{e^-}$-critical.

Corollary 3 *Let $\mathcal{T}$ be an isolate free graph such that $\mathcal{T} \in \mathcal{T}$, then γ^c_{cer}— set of $\mathcal{T}$ will be a star or double star graph.*

Proposition 4 *Let $\mathcal{Z}$ be an isolate free graph of order n and let $\mathcal{C}$ be the γ^c_{cer}— set of $\mathcal{Z}$. For any edge $e = uv \in E_{\mathcal{Z}}$, where $u \in \mathcal{C}$ and $v \notin \mathcal{C}$, if $|N(v) \cap \mathcal{C}| = 1$, then the graph $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical.*

Proof. Let $\mathcal{Z}$ be any isolate free graph of order n and $\mathcal{C}$ be the γ^c_{cer}—set of graph $\mathcal{Z}$. Let $e = uv \in E_{\mathcal{Z}}$ be such that $u \in \mathcal{C}, v \notin \mathcal{C}$ and $|N(v) \cap \mathcal{C}| = 1$. Since $\mathcal{C}$ is γ^c_{cer}—set of the graph $\mathcal{Z}$, therefore $|N(w) \cap (V_{\mathcal{Z}} \backslash \mathcal{C})| = 0$ or $\geq 2, \forall w \in \mathcal{C}$,

that is, every vertex in $\mathcal{C}$ has 0 or at least two neighbors in $V_{\mathcal{Z}}\backslash\mathcal{C}$, implying that $|N(u) \cap (V_{\mathcal{Z}}\backslash\mathcal{C})| \geq 2$, as v is one such neighbor of u in $V_{\mathcal{Z}}\backslash\mathcal{C}$. By assumption, the vertex u cannot be weak support of the graph $\mathcal{Z}$. We consider two cases: $u \in S_2(\mathcal{Z})$ and $u \notin S_2(\mathcal{Z})$.
Case 1. $u \in S_2(\mathcal{Z})$.
If $u \in S_2(\mathcal{Z})$, then $deg_{\mathcal{Z}}(u) \geq 2$, and $v \in L_{\mathcal{Z}}$ is one such neighbor of u in $\mathcal{Z}$. Also, every strong support vertex of a graph $\mathcal{Z}$ belongs to every γ^c_{cer}—set of the graph $\mathcal{Z}$. If we eliminate the edge $e = uv$ from the graph $\mathcal{Z}$ then the CCDN of the graph $\mathcal{Z} - e$ will change, as u is the only neibhor of v in $\mathcal{C}$, implying that $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical.
Case 2. $u \notin S_2(\mathcal{Z})$.
Since $|N(v) \cap \mathcal{C}| = 1$, so if $u \notin S_2(\mathcal{Z})$ then its neighbor v cannot be a leaf in the graph $\mathcal{Z}$, because if v is a leaf, then the vertex $u \in S_2(\mathcal{Z})$ which will be a contradiction to our assumption. As u is the only neighbor of the vertex v in γ^c_{cer}—set $\mathcal{C}$ of the graph $\mathcal{Z}$, therefore if we delete the edge $e = uv$ from the graph $\mathcal{Z}$, then the vertex $v \in \mathcal{Z} - e$ will be the only vertex in the graph $\mathcal{Z} - e$ which is not adjacent to any of the vertex in γ^c_{cer}—set $\mathcal{C}$ of the graph $\mathcal{Z}$, implying that the removal of the edge $e = uv$ from the graph $\mathcal{Z}$ changes the CCDN of $\mathcal{Z} - e$, i.e., $\gamma^c_{cer}(\mathcal{Z} - e) > \gamma^c_{cer}(\mathcal{Z})$ as removal of an edge cannot decrease the CCDN of a graph.
Hence from the above two cases, we conclude that for any edge $e = uv$ such that $u \in \mathcal{C}, v \notin \mathcal{C}$ and $|N(v) \cap \mathcal{C}| = 1$, then $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical. □

Proposition 5 *Let $\mathcal{Z}$ be an isolate free graph of order n, then $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical if there exists a vertex $u \in \mathcal{Z}$ such that $u \in S_2(\mathcal{Z})$.*

Proof. Let $\mathcal{C}$ be the γ^c_{cer}—set of the graph $\mathcal{Z}$. Suppose there exists a vertex $u \in \mathcal{Z}$ such that $u \in S_2(\mathcal{Z})$. Let $v \in L_{\mathcal{Z}}$ be a leaf of the graph $\mathcal{Z}$ adjacent to the strong support vertex u. Since u is the only neighbor of the vertex v in graph $\mathcal{Z}$ and every strong support vertex of a graph $\mathcal{Z}$ belongs to every γ^c_{cer}—set of the graph $\mathcal{Z}$. Therefore $u \in \mathcal{C}, v \in L_{\mathcal{Z}}$ i.e., $v \notin \mathcal{C}$ and $|N(v) \cap \mathcal{C}| = 1$ implies that $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical by preposition 4. □

Theorem 6 *A connected graph $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical if and only if $\mathcal{Z} \in \mathcal{T}$.*

Proof. Suppose that $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical graph. Let $\mathcal{C}$ be the γ^c_{cer}—set of the graph $\mathcal{Z}$. If l is the leaf in the subgraph $\mathcal{Z}[\mathcal{C}]$ induced by γ^c_{cer}— set $\mathcal{C}$, then l is adjacent to a node of degree at least 2 in $\mathcal{Z}[\mathcal{C}]$ and by observation 1 $|epn(l, \mathcal{C})| \geq 1$. Thus, l is a neighbor of at least two nodes in $V_{\mathcal{Z}}\backslash\mathcal{C}$ because l has an external

private neighbor. The fact that $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical is contradicted by the assumption that if $\mathcal{C}$ is γ^c_{cer}—set in $\mathcal{Z}-e$, $\forall e \in \mathcal{Z}$, then $\gamma^c_{cer}(\mathcal{Z}-e) \leq |\mathcal{C}| = \gamma^c_{cer}(\mathcal{Z})$. As a result, the set $\mathcal{C}$ is not a γ^c_{cer}—set in $\mathcal{Z}-e$ $\forall e \in \mathcal{Z}$. Hence $V_{\mathcal{Z}}\backslash\mathcal{C}$ is an independent set and every node in $V_{\mathcal{Z}}\backslash\mathcal{C}$ is adjacent to precisely one node of $\mathcal{C}$ and is thus, a leaf of $\mathcal{Z}$. Since $\mathcal{C}$ is the $\gamma^c_{cer}-$ set of the graph $\mathcal{Z}$, the subgraph $\mathcal{Z}[\mathcal{C}]$ induced by $\mathcal{C}$ in $\mathcal{Z}$ is connected. Hence $\mathcal{Z}[\mathcal{C}]$ will either be a double star or star graph, and so $\mathcal{Z} \in \mathcal{T}$. We may suppose that $\mathcal{Z}[\mathcal{C}]$ is a star $S_{(1,r)}$. As each node in the induced star $\mathcal{Z}[\mathcal{C}]$ is adjacent to at least two nodes in $V_{\mathcal{Z}}\backslash\mathcal{C}$. Let $\mathcal{J}$ denote the set of $q = 2r$, $r \geq 2$ nodes in $V_{\mathcal{Z}}\backslash\mathcal{C}$ that is adjacent to the set of r leaves in $\mathcal{Z}[\mathcal{C}]$, then $\mathcal{Z}[\mathcal{C} \cup \mathcal{J}] = S^*_{(1,r)}$ is a subdivided star graph and can be obtained by adding each node in $\mathcal{C}$ with at least two pendant edges. Thus $\mathcal{Z} \in \mathcal{T}$.
Now, assume that $\mathcal{Z} = (V_{\mathcal{Z}}, E_{\mathcal{Z}}) \in \mathcal{T}$ and let $e = uv \in E_{\mathcal{Z}}$ be any edge in the graph $\mathcal{Z}$. If the edge $e = uv$ is such that one of the end vertex of e is a leaf in $\mathcal{Z}$, then $\gamma^c_{cer}(\mathcal{Z}-e) = \infty$, and so the edge $e = uv$ is critical. Therefore, we may suppose that the edge e is not a pendant edge in $\mathcal{Z}$. More precisely, $\mathcal{Z}$ is not a star graph $S_{(1,r)}$. If $\mathcal{Z}$ is a double-star graph $S_{(q,r)}$ with central vertices u_1 and u_2, then the edge $e = u_1u_2$ joins the two vertices u_1 and u_2 of $\mathcal{Z}$. Thus, $\gamma^c_{cer}(\mathcal{Z}-e) = \infty$ while $\gamma^c_{cer}(\mathcal{Z}) = 2$, so the edge $e = uv$ is critical, and graph $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical. Therefore, let's suppose $\mathcal{Z}$ isn't a double star. Henceforth, $\mathcal{Z}$ is the graph formed from a star $S_{(1,r)}$ for some $r \geq 2$, by appending at least two pendant edges to each leaf of $S_{(1,r)}$. In the set $E_{\mathcal{Z}}\backslash E_{S_{(1,r)}}$ every edge is a pendant edge in $\mathcal{Z}$. Hence, by our previous assumption $e \in E_{S_{(1,r)}}$. But then $\gamma^c_{cer}(\mathcal{Z}-e) = \infty$, which again implies that the edge e is critical. Therefore, $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical. □

Corollary 7 *If $\mathcal{Z}$ is an isolate free graph of order $n \geq 4$, then $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical if $\mathcal{Z}$ has unique minimal γ^c_{cer}—set.*

Observation 2 If $\mathcal{Z} = \mathcal{H} \circ \mathcal{K}$ is the corona of graphs $\mathcal{H}$ and $\mathcal{K}$, then $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical.

3 $[\gamma^c_{cer}]^{e^-}$-stable graphs

We present the characterization of $[\gamma^c_{cer}]^{e^-}$-stable graphs in this section. Note that $\gamma^c_{cer}(\mathcal{Z}) = \infty$ if $\mathcal{Z}$ is a graph containing atleast one isolated vertex. As a result, if we eliminate any pendant edge e from the graph $\mathcal{Z}$, then $\gamma^c_{cer}(\mathcal{Z}-e) = \infty$.

We have observed that if $\delta(\mathcal{Z}) = 1$, then the graph $\mathcal{Z}$ is always $[\gamma^c_{cer}]^{e^-}$ −critical. For a graph $\mathcal{Z}$ to be a $[\gamma^c_{cer}]^{e^-}$-stable graph if $\delta(\mathcal{Z}) \geq 2$. For this purpose, we need the following proposition.

Proposition 8 *Let $\mathcal{Z}$ be an isolate-free graph, then the following two conditions hold:*

(1) If $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-stable, then $\delta(\mathcal{Z}) \geq 2$.

(2) If the edge $e \in E_{\mathcal{Z}}$ is stable, then every $\gamma^c_{cer}(\mathcal{Z}-e)$−set is a $\gamma^c_{cer}(\mathcal{Z})$−set.

Proof.

(1) Suppose, on the contrary, that $\delta(\mathcal{Z}) = 1$. Then graph $\mathcal{Z}$ has at least one pendant edge e incident on a leaf vertex. Since $\mathcal{Z}$ is an isolate-free graph, the removal of the edge e from the graph $\mathcal{Z}$ will result in a graph $\mathcal{Z} - e$ containing an isolated vertex and, therefore $\gamma^c_{cer}(\mathcal{Z} - e) = \infty$, implying that $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical, a contradiction. Hence, $\delta(\mathcal{Z}) \geq 2$ if $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-stable graph.

(2) Suppose $e \in E_{\mathcal{Z}}$ is a stable edge of the graph $\mathcal{Z}$; it means that the removal of the edge $e \in E_{\mathcal{Z}}$ from the graph $\mathcal{Z}$ does not change its CCDN; that is $\gamma^c_{cer}(\mathcal{Z} - e) = \gamma^c_{cer}(\mathcal{Z}) = |\gamma^c_{cer}(\mathcal{Z}) - set|$, which implies that every γ^c_{cer}− set of the graph $\mathcal{Z} - e$ is a γ^c_{cer}−set of the graph $\mathcal{Z}$

□

Proposition 9 *A graph $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-stable if and only if for each $e = uv \in E_{\mathcal{Z}}$ and $\delta(\mathcal{Z}) \geq 2$, $\exists$ a $\gamma^c_{cer}(\mathcal{Z})$-set $\mathcal{C}$ such that:*

(1) $u, v \notin C$.

(2) If $u, v \in \mathcal{Z}$, then $|N_{\mathcal{Z}}(u) \cap \mathcal{C}| \geq 2$ and $|N_{\mathcal{Z}}(v) \cap \mathcal{C}| \geq 2$.

(3) If $u \in \mathcal{C}$ and $v \notin \mathcal{C}$, then $|N_{\mathcal{Z}}(v) \cap \mathcal{C}| \geq 2$.

Proof. Suppose that $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-stable. By preposition 8, $\delta(\mathcal{Z}) \geq 2$. Let $e = uv$ be any edge of the graph $\mathcal{Z}$. Let $\mathcal{Z}' = \mathcal{Z} - uv$ and let $\mathcal{C}$ be any γ^c_{cer}− set of the graph $\mathcal{Z}'$. By proposition 8, the set $\mathcal{C}$ is a γ^c_{cer}−set of the graph $\mathcal{Z}$. Now, if $u, v \in \mathcal{Z}$, then condition (1) holds. Assume that $u, v \in \mathcal{C}$, then since $\mathcal{C}$ is γ^c_{cer}− set of the graph $\mathcal{Z}'$, $|N_{\mathcal{Z}'}(u) \cap \mathcal{C}| \geq 1$ and $|N_{\mathcal{Z}'}(v) \cap \mathcal{C}| \geq 1$, and so $|N_{\mathcal{Z}}(u) \cap \mathcal{C}| \geq 2$ and $|N_{\mathcal{Z}}(v) \cap \mathcal{C}| \geq 2$, thus condition (2) holds. If $u \in \mathcal{C}$

and $v \notin \mathcal{C}$, then since $\mathcal{C}$ is $\gamma^c_{cer}-$ set of the graph $\mathcal{Z}^{'}, |N_{\mathcal{Z}'}(v) \cap \mathcal{C}| \geq 1$, and so $|N_{\mathcal{Z}}(v) \cap \mathcal{C}| \geq 2$. Thus (3) holds. Henceforth, the set $\mathcal{C}$ is a $\gamma^c_{cer}-$ set of the graph $\mathcal{Z}$ such that one of the three conditions (1), (2), and (3) is satisfied.
To prove the sufficiency, assume that $\delta(\mathcal{Z}) \geq 2$, and for every edge, $e = uv \in E_{\mathcal{Z}}$ there exists a $\gamma^c_{cer}-$set $\mathcal{C}$ of the graph $\mathcal{Z}$ satisfying the three conditions $(1), (2)$, and (3). Note that in all three conditions, the set $\mathcal{C}$ is also a $\gamma^c_{cer}-$set for $\mathcal{Z} - uv$. Hence, $\gamma^c_{cer}(\mathcal{Z}) \leq \gamma^c_{cer}(\mathcal{Z} - uv) \leq |\mathcal{C}| = \gamma^c_{cer}(\mathcal{Z})$, implying that $\gamma^c_{cer}(\mathcal{Z}) = \gamma^c_{cer}(\mathcal{Z} - uv)$. Therefore, the graph $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-stable.

□

We have the following observation as a result of proposition 9.
Observation 3. Let $\mathcal{Z}$ be a $[\gamma^c_{cer}]^{e^-}$- stable graph, then $\mathcal{Z}$ has at least two distinct $\gamma^c_{cer}-$sets.
Observation 4.

(a) Every cycle graph C_n is a $[\gamma^c_{cer}]^{e^-}$-stable graph $\forall n \geq 4$.

(b) For every integer $s \geq 4$, $\exists$ s^c_{cer}-stable graph.

Theorem 10 *Let $\mathcal{Z}$ be a complete bipartite graph $K_{(m,n)}, m, n \geq 3$, then $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-stable if and only if $|N_{\mathcal{Z}}(x) \cap T^c_{cer}(\mathcal{Z})| \geq 2$, $\forall x \in \mathcal{Z}$.*

Proof. Assume that $\mathcal{Z}$ is a complete bipartite $[\gamma^c_{cer}]^{e^-}$-stable graph and $x \in V_{\mathcal{Z}}$. Let $\mathcal{C}$ be a $\gamma^c_{cer}-$ set of the graph $\mathcal{Z}$. Then, there exists a vertex $y \in \mathcal{C}$ that is adjacent to x. Now, by definition of the set $T^c_{cer}(\mathcal{Z})$, we note that $\mathcal{C} \subseteq T^c_{cer}(\mathcal{Z})$ and so $y \in N_{\mathcal{Z}}(x) \cap T^c_{cer}(\mathcal{Z})$. Let $C^{'}$ be the $\gamma^c_{cer}-$ set of the graph $\mathcal{Z}^{'} = \mathcal{Z} - uv$, and let z be a vertex in $C^{'}$ adjacent to x. By preposition 8, $C^{'}$ is a$\gamma^c_{cer}-$ set of the graph $\mathcal{Z}$, and so $C^{'} \subseteq T^c_{cer}(\mathcal{Z})$. Thus, $z \in N_{\mathcal{Z}}(x) \in T^c_{cer}(\mathcal{Z})$. Since $y \notin z$, we have $|N_{\mathcal{Z}}(x) \cap T^c_{cer}(\mathcal{Z})| \geq |\{y, z\}| = 2$, as claimed.
For sufficiency, let $\mathcal{Z}$ be a complete bipartite graph $K_{(m,n)}, m, n \geq 3$, and $\mathcal{C}$ be the $\gamma^c_{cer}-$ set of the graph $\mathcal{Z}$. Suppose that $|N_{\mathcal{Z}}(x) \cap T^c_{cer}(\mathcal{Z})| \geq 2, \forall x \in \mathcal{Z}$. Let $y \in \mathcal{C}$ be a vertex in the $\gamma^c_{cer}-$ set of the graph $\mathcal{Z}$ adjacent to the vertex x. Then in the graph $\mathcal{Z}^{'} = \mathcal{Z} - uv$, $|N_{\mathcal{Z}'}(x) \cap T^c_{cer}(\mathcal{Z})| \geq 1$, and so $|N_{(}\mathcal{Z}^{'})(x) \cap \mathcal{C}| \geq 1$, implying that $|N_{\mathcal{Z}}(x) \cap \mathcal{C}| \geq 2$. Hence by proposition 9, $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-stable.

□

We have observed that the above theorem is not true for bipartite graphs, and its converse does not hold for graphs in general. See the examples below for demonstration.
Example 1. Let $\mathcal{Z}$ be a bipartite graph shown in figure 2 below. The colored vertex set u, v form the $\gamma^c_{cer}-$ set of the graph $\mathcal{Z}$, implying that $\gamma^c_{cer}(\mathcal{Z}) = 2$.

However, $\gamma^c_{cer}(\mathcal{Z}' = \mathcal{Z} - uv) = 6$, which implies that $\gamma^c_{cer}(\mathcal{Z}) \neq \gamma^c_{cer}(\mathcal{Z}')$ and hence $\mathcal{Z}$ is not $[\gamma^c_{cer}]^{e^-}$-stable graph.

Example 2. If $\mathcal{Z}$ is a graph obtained from a 6−cycle $x_1x_2...,x_6x_1$ by adding the chord x_1x_4 and x_3x_6 then $\mathcal{Z}$ has exactly two γ^c_{cer}−sets, namely the sets x_1, x_4 and x_3, x_6. Thus $T^c_{cer}(\mathcal{Z}) = \{x_1, x_4, x_3, x_6\}$ and $|N_{\mathcal{Z}}(x) \cap T^c_{cer}(\mathcal{Z})| \geq 2, \forall x \in \mathcal{Z}$. However the edges x_1x_4 and x_3x_6, are both critical in $\mathcal{Z}$, and so $\mathcal{Z}$ is not a $[\gamma^c_{cer}]^{e^-}$-stable graph.

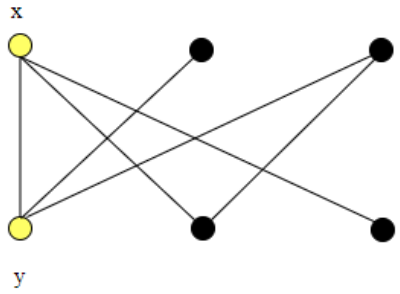


Figure 2: Bipartite graph $K_{3,3}$.

Consequently, as a direct conclusion of Theorem 10, we have the following result.

Corollary 11 *Let $\mathcal{Z}$ be a bipartite graph such that $\mathcal{Z}$ has two disjoint γ^c_{cer}−sets then $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-stable.*

Next, we will show that if a graph $\mathcal{Z}$ has at least two disjoint γ^c_{cer}− sets then $\mathcal{Z}$ cannot have critical edges more then $\gamma^c_{cer}(\mathcal{Z})$

Proposition 12 *If $\mathcal{Z}$ is connected graph of order n such that $\mathcal{Z}$ has two disjoint γ^c_{cer}−sets, then $\mathcal{Z}$ has a maximum of $\gamma^c_{cer}(\mathcal{Z})$ critical edges.*

Proof. Suppose that $\mathcal{X}$ and $\mathcal{Y}$ be two disjoint γ^c_{cer}−sets of an isolate free graph $\mathcal{Z}$ of order n. If the graph $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-stable, then every edge of $\mathcal{Z}$ stable, and the result is thus straightforward. Assume that $\mathcal{Z}$ contains at least one $[\gamma^c_{cer}]^{e^-}$-critical edge and define $e = uv$ as such an edge. If e has no end in the set $\mathcal{X}$, then the set $\mathcal{X}$ is a γ^c_{cer}− set in $\mathcal{Z} - e$, which implies that $\gamma^c_{cer}(\mathcal{Z} - e) = \gamma^c_{cer}(\mathcal{Z})$, a contradiction. Therefore, e has at least one end in $\mathcal{X}$. Likewise, e has at least one end in $\mathcal{Y}$. Thus, $e = uv$ where $u \in \mathcal{X}$ and $v \in \mathcal{Y}$. If $|N(v) \cap \mathcal{X}| \geq 2$, then $\mathcal{X}$ is a γ^c_{cer}− set in $\mathcal{Z} - e$, a contradiction. Hence, $N(v) \cap \mathcal{X} = \{u\}$, and similarly, $N(u) \cap \mathcal{Y} = \{v\}$. This means that $\mathcal{Z}$ has critical edges that are at most the CCDN of the graph $\mathcal{Z}$. □

Corollary 13 *For any integer* $t \geq 1, \exists$ *a graph* $\mathcal{Z}$ *with precisely two disjoint* $\gamma^c_{cer}-$ *sets such that* $\gamma^c_{cer}(\mathcal{Z}) = t$ *and* $\mathcal{Z}$ *has precisely* t *critical edges.*

Observation 5. The following graphs are $[\gamma^c_{cer}]^{e^-}$-stable.

(1) Complete graph K_n is $[\gamma^c_{cer}]^{e^-}$-stable for all $n \geq 3$.

(2) Complete bipartite graph $K_{(m,n)}$ is $[\gamma^c_{cer}]^{e^-}$-stable for all $m, n \geq 3$.

(3) Bipartite graphs satisfying the corollary 11 are $[\gamma^c_{cer}]^{e^-}$-stable.

(4) Cycle graph C_n is $[\gamma^c_{cer}]^{e^-}$-stable for all $n \geq 4$.

4 Conclusion

The study of criticality and stability of graphs upon edge removal or addition on any graph domination parameter has exciting applications in networking. In this article, we have initiated the study of connected certified domination criticality and stability upon edge removal.

For connected certified domination edge critical graphs, we have proved that every graph with $dia(\mathcal{Z}) \leq 2$ and $\delta(\mathcal{Z} = 1$,is $[\gamma^c_{cer}]^{e^-}$-critical, and if $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical graph with $dia(\mathcal{Z}) \leq 2$, then for every $\gamma^c_{cer}(\mathcal{Z})-$ set $\mathcal{C}$ of $\mathcal{Z}$, $\mathcal{Z}[\mathcal{C}]$ is either a trivial graph or a star graph. Also, if $\mathcal{C}$ is the $\gamma^c_{cer}-$ set of a graph $\mathcal{Z}$ and for any edge $e = uv \in \mathcal{Z}$, where $u \in \mathcal{C}$ and $v \notin C$, if $|N(v) \cap \mathcal{C}| = 1$, then the graph $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical, and $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical if the graph $\mathcal{Z}$ contains a support vertex. We have proved a necessary and sufficient condition that a graph $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-critical iff $\mathcal{Z} \in \mathcal{T}$.

Similarly, for connected certified domination edge stable graphs, we have proved the following results:

If a graph $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-stable, then $\delta(\mathcal{Z}) \geq 2$, and if the edge $e = uv \in E_{\mathcal{Z}}$ is stable, then every $\gamma^c_{cer}(\mathcal{Z}-e)-$ set is a $\gamma^c_{cer}(\mathcal{Z})-$set. Also, a graph $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-stable if and only if for each $e = uv \in E_{\mathcal{Z}}$ and $\delta(\mathcal{Z}) \geq 2, \exists$ a $\gamma^c_{cer}(\mathcal{Z})-$set $\mathcal{C}$ such that: (1). $u, v \notin \mathcal{C}$. (2). If $u, v \in \mathcal{C}$, then $|N_{\mathcal{Z}}(u) \cap \mathcal{C}| \geq 2$ and $|N_{\mathcal{Z}}(v) \cap \mathcal{C}| \geq 2$. (3). If $u \in \mathcal{C}$ and $v \notin \mathcal{C}$, then $|N_{\mathcal{Z}}(v) \cap \mathcal{C}| \geq 2$. We have shown that if $\mathcal{Z}$ is a complete bipartite graph $K_{(m,n)}, m, n \geq 3$, then $\mathcal{Z}$ is $[\gamma^c_{cer}]^{e^-}$-stable if and only if $|N_{\mathcal{Z}}(x) \cap T^c_{cer}(\mathcal{Z})| \geq 2, \forall x \in \mathcal{Z}$, and we have justified that this result is not true for bipartite graphs, and its converse is not valid for graphs in general. And finally, we have shown that if $\mathcal{Z}$ is a connected graph of order n such that G has precisely two disjoint $\gamma^c_{cer}-$sets, then $\mathcal{Z}$ has a maximum of $\gamma^c_{cer}(\mathcal{Z})$ critical edges.

References

[1] S. B. Akers, D. Harel, B. Krishnamurthy, The star graph: An attractive alternative to the n-cube, *Interconnection networks for high-performance parallel computers*, 1994. pp. 145–152. ⇒28

[2] V. Anithakumari, M. Padmini, Total domination number in graphs and graph modification, *Tamilnadu: International Journal of Mathematics Trends and Technology (IJMTI)*, (2020). ⇒26

[3] F. T. Boesch, S. Chen,J. A. M. McHugh, On covering the points of a graph with point disjoint paths, *In Graphs and Combinatorics:Proceedings of the Capital Conference on Graph Theory and Combinatorics at the George Washington University,* Springer Berlin Heidelberg (1974) 201–212. ⇒28

[4] M. Chellali, N. Jafari Rad, S. M. Sheikholeslami, L. Volkmann, Roman domination in graphs, In: *Topics in Domination in Graphs* Vol. 64. Springer, Cham. 2020, pp. 365–409 ⇒26

[5] X. G. Chen, L. Sun, D. Xiang Ma, Connected domination critical graphs, *Applied Mathematics Letter* **17,** 5 (2004) 503–507. ⇒26

[6] W. J. Desormeaux, T. W Haynes, M. A. Henning, Total domination critical and stable graphs upon edge removal, *Discrete applied mathematics* **158,** 15 (2010) 1587–1592. ⇒26

[7] W. J. Desormeaux, T. W. Haynes, L. van der Merwe, Connected domination stable graphs upon edge addition, *Quaestiones Mathematicae* **38,** 6 (2015) 841–848. ⇒26

[8] M. Dettlaff, M. Lemanska, J. Topp, R. l. Ziemann, P. Zylinski, Certified domination, *AKCE International Journal of Graphs and Combinatorics* **17,** 1 (2020) 86–97. ⇒25, 26

[9] M. Dettlaff, M. Lemanska, J. Topp, R. l. Ziemann, P. Zylinski, Graphs with equal domination and certified domination numbers, *arXiv:1710.02059* (2017). ⇒26

[10] R. D. Dutton, R. C. Brigham, An extremal problem for edge domination insensitive graphs, *Discrete applied mathematics* **20,** 2 (1988) 113–125. ⇒26

[11] J. A. Gallian, A dynamic survey of graph labeling, *Electronic Journal of combinatorics*, 1 (DynamicSurveys) (2018). ⇒28

[12] E. Harris, *Global Domination Stable Graphs*, PhD thesis, East Tennessee State University, 2012. ⇒26

[13] T. W. Haynes, S. Hedetniemi, P. Slater, *Fundamentals of Domination in Graphs,* CRC press, 2013. ⇒27

[14] P. K. Jakkepalli, S. Arumugam, H. Khandelwal, Algorithmic aspects of certified domination in graphs, *Communications in Combinatorics and Optimization* **7,** 2 (2022) 247–255. ⇒26

[15] S. D. Raj, S. G. Shiji Kumari, Certified domination number in product of graphs, *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* **11,** 3 (2020) 1166–1170. ⇒26

[16] S. D. Raj, S. G. Shiji Kumari, A. M. Anto, Certified domination number in corona product of graphs, *Malaya Journal of Matematik* **9,** 1 (2021) 1080–1082. ⇒26

[17] H. B, Walikar, B. D. Acharya, *Domination Critical Graphs,* National Academy Science Letters-India **2,** 2 (1979) 70–72. ⇒26

[18] D. B. West, *Introduction to Graph Theory. Vol. 2.* Prentice Hall, Upper Saddle River, 2001. ⇒27

Received: January 30, 2023 • Revised: April 17, 2023

Acta Univ. Sapientiae Informatica **15,** 1 (2023) 38–45

DOI: 10.2478/ausi-2023-0004

On the spread of the distance signless Laplacian matrix of a graph

S. PIRZADA
Department of Mathematics,
University of Kashmir, Srinagar,
India
email:
pirzadasd@kashmiruniversity.ac.in

Mohd Abrar Ul HAQ
Department of Mathematics,
University of Kashmir, Srinagar,
India
email: abrar14789@gmail.com

Abstract. Let G be a connected graph with n vertices, m edges. The distance signless Laplacian matrix $D^Q(G)$ is defined as $D^Q(G) = \mathrm{Diag}(\mathrm{Tr}(G)) + D(G)$, where $\mathrm{Diag}(\mathrm{Tr}(G))$ is the diagonal matrix of vertex transmissions and $D(G)$ is the distance matrix of G. The distance signless Laplacian eigenvalues of G are the eigenvalues of $D^Q(G)$ and are denoted by $\partial_1^Q(G), \partial_2^Q(G), \ldots, \partial_n^Q(G)$. ∂_1^Q is called the distance signless Laplacian spectral radius of $D^Q(G)$. In this paper, we obtain upper and lower bounds for $S_{D^Q}(G)$ in terms of the Wiener index, the transmission degree and the order of the graph.

Key words and phrases: distance matrix; distance signless Laplacian matrix; distance signless Laplacian eigenvalues; spread; Wiener index; transmission degree

1 Introduction

Let G be a connected simple graph with vertex set $V(G) = \{v_1, v_2, \ldots, v_n\}$) and edges set $E(G)$. In G, the distance $d(v_i, v_j)$ between the vertices v_i and v_j is the length of (number of edges) the shortest path that connects v_i and v_j. The diameter of G is the maximum distance between any two vertices of G. The distance matrix of G is an $n \times n$ matrix in which the $(i, j)^{th}$-entry is equal to the distance between vertices v_i and v_j, that is, $D_{i,j}(G) = d_{i,j} = d(v_i, v_j)$. For more definitions and notations, we refer to [10].

In G, the distance degree of a vertex v, denoted by $Tr_G(v)$, is defined to be the sum of the distances from v to all other vertices in G, that is, $Tr_G(v) = \sum_{u \in V(G)} d(u, v)$. We can also write $Tr_G(v_i)$ as Tr_i. A graph G is said to be k-transmission regular if $Tr_i = k$, for each $i = 1, 2, \ldots, n$. The transmission degree sequence is given by $\{Tr_1, Tr_2, Tr_3, \ldots, Tr_n\}$. The second transmission degree of v_i, denoted by T_i, is given by $T_i = \sum_{j=1}^{n} d_{ij} Tr_j$. The Wiener index of graph G, denoted by $W(G)$, is the sum of the distances between all unordered pairs of vertices in G, that is,

$$W(G) = \frac{1}{2} \sum_{u,v \in V(G)} d(u, v) = \frac{1}{2} \sum_{u \in V(G)} Tr_G(v).$$

Let $Tr_G = diag(Tr_1, Tr_2, \ldots, Tr_n)$ be the diagonal matrix of vertex transmissions of G. Aouchiche and Hansen [5] introduced the Laplacian and the signless Laplacian for the distance matrix of a connected graph G. The matrix $DQ(G) = Tr(G) + D(G)$ (or simply D^Q) is called the distance signless Laplacian matrix of G. Since $DQ(G)$ is symmetric (positive semi-definite), its eigenvalues can be arranged as: $\partial_1^Q(G) \geq \partial_2^Q(G) \geq \cdots \geq \partial_n^Q(G)$, where $\partial_1^Q(G)$ is called the distance signless Laplacian spectral radius of G. If $\partial_i^Q(G)$ is repeated p times, then we say that the multiplicity of $\partial_i^Q(G)$ is p and we write $m(\partial_i^Q(G)) = p$. As $D^Q(G)$ is nonnegative and irreducible, by the Perron-Frobenius theorem, $\partial_1^Q(G)$ is positive, simple and there is a unique positive unit eigenvector X corresponding to $\partial_1^Q(G)$, which is called the distance signless Laplacian Perron vector of G. The distance signless Laplacian spread of a graph G, denoted by $S_{DQ}(G)$, is defined as $S_{DQ}(G) = \partial_1^Q(G) - \partial_n^Q(G)$, where $\partial_1^Q(G)$ and $\partial_n^Q(G)$ are respectively the largest and the smallest distance signless Laplacian eigenvalues

of G. Some recent work on distance signless Laplacian eigenvalues can be seen in [1, 4, 8, 11, 12, 13, 14].

The rest of the paper is organized as follows. In Section 2, we obtain lower and upper bounds for $S_{DQ}(G)$ in terms of the Wiener index $W(G)$, the transmission Tr and the order n of G.

2 Bounds for spread of distance signless Laplacian matrix

For a graph G with n vertices, let $Tr_{max}(G) = max\{Tr(v) : v \in V(G)\}$ and $Tr_{min}(G) = min\{Tr(v) : v \in V(G)\}$. Whenever the graph G is understood, we will write Tr_{max} and Tr_{min} in place of $Tr_{max}(G)$ and $Tr_{min}(G)$, respectively. From the definitions, we have $2W(G) = \partial_1^Q + \partial_2^Q + \cdots + \partial_n^Q$. Also, $Tr_{max} \geq \frac{2W(G)}{n}$ and $Tr_{min} \leq \frac{2W(G)}{n}$, where $\frac{2W(G)}{n}$ is the average transmission degree. First we note the following observations.

Lemma 1 [2] *Let G be a simple, connected graph. Then*

$$\frac{Tr_{min} + \sqrt{Tr_{min}^2 + 8T_{min}}}{2} \leq \partial_1^Q(G) \leq \frac{Tr_{max} + \sqrt{Tr_{max}^2 + 8T_{max}}}{2},$$

equality hold if and only if the graph is transmission regular.

Lemma 2 [6] *Let G be a connected graph with minimum and maximum transmissions Tr_{min} and Tr_{max}. Then $2Tr_{min} \leq \partial_1^Q(G) \leq 2Tr_{max}$, and the equality hold if and only if G is transmission regular.*

Now, we obtain bounds for the distance signless Laplacian spread $S_{DQ}(G)$ of a graph G in terms of the Wiener index $W(G)$, the order n, the maximum transmission degree $Tr_{max}(G)$ and the minimum transmission degree Tr_{min} of G.

Theorem 3 *Let G be a connected graph with n vertices having Wiener index $W(G)$. Then*

$$\frac{n(Tr_{min} + \sqrt{Tr_{min}^2 + 8T_{min}}) - 4W(G)}{2(n-1)} \leq S_{DQ}(G)$$
$$< \frac{n(Tr_{max} + \sqrt{Tr_{max}^2 + 8T_{min}}) - 4W(G)}{2}.$$

Equality holds in the left if and only if $G \cong K_n$.

Proof. Let $\partial_1^Q(G), \partial_2^Q(G), \ldots, \partial_n^Q(G)$ be $D^Q(G)$-eigenvalues. Then we have

$$2W(G) = \partial_1^Q(G) + \partial_2^Q(G) + \cdots + \partial_n^Q(G) \geq \partial_1^Q(G) + (n-1)\partial_n^Q(G),$$

which implies that $\partial_n^Q(G) \leq \frac{2W(G)-\partial_1^Q(G)}{n-1}$, with equality if and only if $\partial_2^Q(G) = \partial_3^Q(G) = \cdots = \partial_n^Q(G)$. For equality, consider the following two cases.
Case 1. Clearly, $\partial_1^Q(G) = \partial_2^Q(G) = \partial_3^Q(G) = \cdots = \partial_n^Q(G)$, is not possible, since the spectral radius of D^Q is always simple.
Case 2. $\partial_1^Q(G) > \partial_2^Q(G)$ and $\partial_2^Q(G) = \partial_3^Q(G) = \cdots = \partial_n^Q(G)$. Then $G \cong K_n$, as K_n is the unique graph having only two distinct distance signless Laplacian eigenvalues. Therefore,

$$\begin{aligned} S_{D^Q}(G) = \partial_1^Q(G) - \partial_n^Q(G) &\geq \partial_1^Q(G) - \frac{2W(G) - \partial_1^Q(G)}{n-1} \\ &= \frac{(n-1)\partial_1^Q(G) - 2W(G) - \partial_1^Q(G)}{n-1} \\ &= \frac{n\partial_1^Q(G) - 2W(G)}{n-1}. \end{aligned}$$

Now, using Lemma 1, we get

$$\begin{aligned} S_{D^Q}(G) &\geq \frac{n(\frac{Tr_{min}+\sqrt{Tr_{min}^2+8T_{min}}}{2}) - 2W(G)}{n-1} \\ &= \frac{n(Tr_{min} + \sqrt{Tr_{min}^2 + 8T_{min}}) - 4W(G)}{2(n-1)}, \end{aligned}$$

with equality if and only if $G \cong K_n$. Also, we have $2W(G) = \partial_1^Q(G) + \partial_2^Q(G) + \cdots + \partial_n^Q(G) \leq (n-1)\partial_1^Q(G) + \partial_n^Q(G)$. We observe that the above inequality is strict as the distance signless Laplacian spectral radius is always simple, that is, $\partial_n^Q(G) \geq 2W(G) - (n-1)\partial_1^Q(G)$. Therefore,

$$S_{D^Q}(G) = \partial_1^Q(G) - \partial_n^Q(G) < \partial_1^Q(G) - 2W(G) + (n-1)\partial_1^Q(G).$$

By using Lemma 1, we get

$$\begin{aligned} S_{D^Q}(G) &\leq \frac{n(Tr_{max} + \sqrt{Tr_{max}^2 + 8T_{min}})}{2} - 2W(G) \\ &= \frac{n(Tr_{max} + \sqrt{Tr_{max}^2 + 8T_{max}}) - 4W(G)}{2} \end{aligned}$$

and we get the desired result. □

The following lemma will be used in the next theorem.

Lemma 4 [15] *Let G be a connected graph on n vertices. Then $\partial_1^Q(G) \geq \frac{4W(G)}{n}$ with equality holding if and only if G is transmission regular.*

Theorem 5 *Let G be a connected graph of order n. Then $S_{D^Q}(G) \geq \frac{2W(G)}{n-1}$, and equality holds if and only if $G \cong K_n$.*

Proof. If $\partial_1^Q(G), \partial_2^Q(G), \ldots, \partial_n^Q(G)$ are $D^Q(G)$-eigenvalues, then we have

$$2W(G) = \partial_1^Q(G) + \partial_2^Q(G) + \cdots + \partial_n^Q(G) \geq \partial_1^Q(G) + (n-1)\partial_n^Q(G),$$

which implies that $\partial_n^Q \leq \frac{2W(G) - \partial_1^Q(G)}{n-1}$, with equality if and only if $G \cong K_n$. Therefore,

$$\begin{aligned} S_{D^Q}(G) = \partial_1^Q(G) - \partial_n^Q(G) &\geq \partial_1^Q(G) - \frac{2W(G) - \partial_1^Q(G)}{n-1} \\ &= \frac{(n-1)\partial_1^Q(G) - 2W(G) + \partial_1^Q(G)}{n-1} \\ &= \frac{n\partial_1^Q(G) - 2W(G)}{n-1} \end{aligned}$$

Using Lemma 4, we get $S_{D^Q}(G) = \partial_1^Q(G) - \partial_n^Q(G) \geq \frac{2W(G)}{n-1}$, equality holds if and only if $G \cong K_n$. □

Since $D^Q(G)$ is nonnegative and irreducible, by the Perron-Frobenius theorem, ∂_1^Q is positive, simple and there is a unique positive unit eigenvector X corresponding to ∂_1^Q. Using Lemma 4 and the fact that $\partial_1^Q(G) \geq$

$\frac{2\sqrt{\sum_{i=1}^{n} Tr_i^2}}{n}$, equality hold if and only if G is transmission degree regular graph [9], we get

$$S_{D^Q}(G) \geq \frac{2(n-1)\sqrt{\sum_{i=1}^{n} Tr_i^2} - 2W(G)}{n-1},$$

and equality holds if and only if G is transmission degree regular graph.

Lemma 6 [3] *If the transmission degree sequence of* G *is* $\{Tr_1, Tr_2 \ldots, Tr_n\}$, *then*

$$\sum_{i=1}^{n} \partial_i^Q(G)^2 = 2 \sum_{1 \leq i < j \leq n} (d_{ij})^2 + \sum_{i=1}^{n} Tr_i{}^2.$$

Theorem 7 *Let G be a connected graph with n vertices. Then*

$$S_{D^Q}(G) \geq 2Tr_{min} - \sqrt{\frac{R_1 - 4Tr_{min}^2}{n-1}},$$

and equality holds if and only if $G \cong K_n$.

Proof. From Lemma 6, we have $\sum_{i=1}^{n} \partial_i^Q(G)^2 = 2\sum_{1 \leq i < j \leq n}(d_{ij})^2 + \sum_{i=1}^{n} Tr_i{}^2 = R_1$. Clearly, $R_1 = \sum_{i=1}^{n} \partial_i^Q(G)^2 \geq \partial_1^Q(G)^2 + (n-1)\partial_n^Q(G)^2$, which implies that $\partial_n^Q(G) \leq \sqrt{\frac{R_1 - \partial_1^Q(G)^2}{n-1}}$, with equality if and only if $G \cong K_n$. By using this inequality for $\partial_n^Q(G)$, we have

$$S_{D^Q}(G) = \partial_1^Q(G) - \partial_n^Q(G) \geq \partial_1^Q(G) - \sqrt{\frac{R_1 - \partial_1^Q(G)^2}{n-1}}$$

Now, using Lemma 2, we get

$$S_{D^Q}(G) \geq 2Tr_{min} - \sqrt{\frac{R_1 - 4Tr_{min}^2}{n-1}},$$

which is the required inequality. Clearly, the equality holds if and only if $G \cong K_n$. □

Remarks. If G is a connected graph of order n, then $\partial_n^Q(G) \leq Tr_{min}$, where Tr_{min} is the smallest transmission [7]. From Theorem 7, we have $S_{DQ}(G) \geq 2Tr_{min} - \partial_n^Q(G)$. Combining, we get $\partial_1^Q(G) - \partial_n^Q(G) \geq Tr_{min}$.

If G is a connected graph of order $n > 2$, then $\partial_1^Q(G) \geq 2(n-1)$ [9]. Using the inequality $\partial_n^Q(G) \leq \frac{2W(G)}{n}$., we get $S_{DQ}(G) = \partial_1^Q(G) - \partial_n^Q(G) \geq 2(n-1) - \frac{2W(G))}{n} = \frac{2(n(n-1)-W(G))}{n}$.

References

[1] A. Alhevaz, M. Baghipur, S. Pirzada, Y. Shang, Some inequalities involving the distance signless Laplacian eigenvalues of graphs, *Transactions Combinatorics* **10(1)** (2021) 9–29. ⇒40

[2] A. Alhevaz, M. Baghipur, E. Hashemi, H. Ramane, On the distance signless Laplacian spectrum of graphs, *Bull. Malays. Math. Sci. Soc.* **42** (2019) 2603–2621. ⇒40

[3] A. Alhevaz, M. Baghipur, S. Paul, On the distance signless Laplacian spectral radius and the distance signless Laplacian energy of graphs, *Discrete Math. Algorithms and Appl.* **10(3)** (2018) 1850035 (19 pages). ⇒43

[4] A. Alhevaz, M. Baghipur, H. Ahmad, S. Pirzada, Brouwer type conjecture for the eigenvalues of distance signless Laplacian matrix of a graph, *Linear Multilinear Algebra* **69** (2019) 1–18. ⇒40

[5] M. Aouchiche, P. Hansen, Two Laplacians for the distance matrix of a graph, *Linear Algebra Appl.* **439** (2013) 21–33. ⇒39

[6] M. Aouchiche, P. Hansen, A signless Laplacian for the distance matrix of a graph, Cahiers du GERAD G-2011-78. ⇒40

[7] K. C. Das, H. Lin, J. Guo, Distance signless Laplacian eigenvalues of graphs, *Front. Math. China* **14** (2019) 693–713. ⇒44

[8] S. Khan, S. Pirzada, Distance signless Laplacian eigenvalues, diameter, and clique number, *Discrete Mathematics Letters* **10** (2022) 28–31. ⇒40

[9] L. Medina, H. Nina, M. Trigo, On distance signless Laplacian spectral radius and distance signless Laplacian energy, *Mathematics* **8** (2020) 792. ⇒43, 44

[10] S. Pirzada, *An Introduction to Graph Theory*, Universities Press, Hyderabad, India, 2012. ⇒39

[11] S. Pirzada, B. A. Rather, M. Aijaz, T. A. Chishti, On distance signless Laplacian spectrum of graphs and spectrum of zero divisor graphs of $\mathbb{Z}_n$, *Linear Multilinear Algebra* **70** (2022) 3354–3369. ⇒40

[12] S. Pirzada, B. A. Rather, R. U. Shaban, M. I. Bhat, On distance Laplacian (signless) eigenvalues of commuting graphs of dihedral and dicyclic groups, *Springer Proceedings on Algebra and Related Topics with Applications*, ICARTA 2019 (2022) 413–425. ⇒40

[13] B. A. Rather, S. Pirzada, T. A. Naikoo, On distance signless Laplacian spectra of power graphs of the integer modulo group, *Art Discrete Appl. Math.* **5** (2022) P2.09. ⇒40

[14] R. U. Shaban, B. A. Rather, S. Pirzada, A. Somasundaram, On distance signless Laplacian spectral radius of power graphs of cyclic and dihedral groups, *Annales Mathematicae et Informaticae* **55** (2022) 172–183. ⇒40

[15] R. Xing, B. Zhou, J. Li, On the distance signless Laplacian spectral radius of graphs, *Linear Multilinear Algebra* **62** (2014) 1377–1387. ⇒42

Received: April 19, 2023 • *Revised: May 15, 2023*

Acta Univ. Sapientiae Informatica **15,** 1 (2023) 46–59

DOI: 10.2478/ausi-2023-0005

Some relations between energy and Seidel energy of a graph

Harishchandra S. RAMANE
Department of Mathematics
Karnatak University
Dharwad, India
email: hsramane@yahoo.com

K. ASHOKA
Department of Mathematics
Christ University
Bangalore, India
email: ashokagonal@gmail.com

B. PARVATHALU
Department of Mathematics
Karnatak University's
Karnatak Arts College
Dharwad, India
email: bparvathalu@gmail.com

Daneshwari PATIL
Department of Mathematics
Karnatak University
Dharwad, India
email: daneshwarip@gmail.com

Abstract. The energy $E(G)$ of a graph G is the sum of the absolute values of eigenvalues of G and the Seidel energy $E_S(G)$ is the sum of the absolute values of eigenvalues of the Seidel matrix S of G. In this paper, some relations between the energy and Seidel energy of a graph in terms of different graph parameters are presented. Also, the inertia relations between the graph eigenvalue and Seidel eigenvalue of a graph are given. The results in this paper generalize some of the existing results.

1 Introduction

Let G be a simple, finite and undirected graph of order n with vertex set $V = \{v_1, v_2 \ldots, v_n\}$. The adjacency matrix $A = [a_{ij}]$ of G is a square matrix of

Key words and phrases: graph eigenvalues, graph energy, Seidel eigenvalues, Seidel energy

order n whose (i,j)-th entry $a_{ij} = 1$ if v_i and v_j are adjacent and 0 otherwise. The complement of a graph G is denoted by $\overline{G}$. The degree of a vertex v_i, denoted by $d(v_i)$, is the number of edges incident with v_i. A graph G is called r- regular if $d(v_i) = r$ for all $v_i \in V$. Let Δ be the maximum degree of G. Much like adjacency matrix, in 1966 J. H. van Lint and J. J. Seidel [24] introduced one more real symmetric $\{0, \pm 1\}$-matrix, called the Seidel matrix S as a tool for studying the systems of equiangular lines in Euclidean space. Later in 1968 J. J. Seidel studied the Seidel eigenvalues of strongly regular graphs [21]. The eigenvalues $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_n$ of the adjacency matrix A are called the eigenvalues of G. A graph is integral if its eigenvalues are integers. Similarly, the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of the Seidel matrix S of G are called the Seidel eigenvalues of G. For a given interval $\mathbf{I}$, $n_\theta(\mathbf{I})$ denotes the number of eigenvalues of G which belongs to the interval $\mathbf{I}$. The number of positive, negative and zero eigenvalues of G are denoted by n^+, n^- and n^0 respectively, called inertia of G.

The graph energy defined by I. Gutman in 1978 [7] and gained its own importance in the spectral graph theory. The energy of a graph G is defined as

$$E(G) = \sum_{j=1}^{n} |\theta_j|.$$

The graph energy has applications in chemistry [6, 12]. An equivalent definition to the energy of a graph G is as follows:

$$E(G) = 2\sum_{j=1}^{n^+} \theta_j = -2\sum_{j=1}^{n^-} \theta_{n-j+1} = 2\max_{1\leq i\leq n} \sum_{j=1}^{i} \theta_j = 2\max_{1\leq i\leq n} \sum_{j=1}^{i} -\theta_{n-j+1}.$$

Two graphs G_1 and G_2 of same order are said be equienergetic if $E(G_1) = E(G_2)$. Similarly, the Seidel energy $E_S(G)$ [8] of a graph G is defined as the sum of the absolute values of eigenvalues of Seidel matrix S. The Seidel energy is invariant under Seidel switching and complement of a graph. The Seidel energy $E_S(G)$ of a graph G introduced by W. H. Haemers in 2012 and presented a relation between energy of a complete graph and Seidel energy of G. However, the exact relation of Seidel energy of graph and bounds for it haven't been much studied in the literature so far. One of the interesting problem on graph energy is to characterize those graphs which are equienergetic with respect to both adjacency and other matrices like distance matrix, Seidel matrix etc. A weaker problem is to construct the families of graphs which are equienergetic with respect to both the adjacency and the other matrices related to graphs. For instance, see [11, 16]. This motivates us to study some

relations between the energy and the Seidel energy in terms of different graph parameters. The research related to the Seidel energy and its variants, see [2, 13, 14, 15, 18, 22, 23]. Two graphs G_1 and G_2 of same order are said be Seidel equienergetic if $E_S(G_1) = E_S(G_2)$. Let K_n and K_{n_1,n_2} $(n = n_1 + n_2)$ denote the complete graph and the complete bipartite graph of order n respectively. This paper is organized as follows. In section 2, basic definitions, known results on eigenvalues, energy, Seidel eigenvalues and Seidel energy of graph are presented. In section 3, the exact relations between the Seidel energy and energy of a regular graph in terms of other graph parameters are given. Also, a large class of Seidel equienergetic graphs are presented. The obtained results in this section generalize some of the existing results. Section 4 provides inertia relations between the graph eigenvalues and Seidel eigenvalues. Also, the relations between the Seidel energy and energy of a graph in terms of other graph parameters are given. As a consequence, some bounds for the Seidel energy of a graph are obtained.

2 Preliminaries

Definition 1 [9] *The line graph* $L(G)$ *of a graph* G *is the graph with vertex set same as the edge set of* G*. In the line graph* $L(G)$ *any two vertices are adjacent if the corresponding edges in* G *have a common vertex. The* k^{th} *iterated line graph of* G *for* $k \in \{0, 1, 2, \ldots\}$ *is defined as* $L^k(G) = L(L^{k-1}(G))$*, where* $L^0(G) = G$ *and* $L^1(G) = L(G)$*.*

Theorem 2 [3] *If* G *is an* r*-regular graph of order* n *with the eigenvalues* $r, \theta_2, \ldots, \theta_n$*, then the eigenvalues of* S *are* $n-2r-1, -1-2\theta_2, \ldots, -1-2\theta_n$*.*

Lemma 3 [10] *Let* P *and* Q *be two Hermitian matrices of same order* n *and* $R = P + Q$*. Then*

$$\mu_{n-i-k}(R) \geq \mu_{n-i}(P) + \mu_{n-k}(Q)$$
$$\mu_{s+t+1}(R) \leq \mu_{s+1}(P) + \mu_{t+1}(Q)$$

where $0 \leq i, k, s, i+k+1, s+t+1 \leq n$ *and* $\mu_j(B)$ *is the* j^{th} *largest eigenvalue of a Hermitian matrix* B*.*

Theorem 4 [1] *Let* G *be a graph of order* n*. Then*

$$E(G) + E(\overline{G}) - 2(n-1) < E_S(G) \leq E(G) + E(\overline{G}).$$

The equality in the right side holds if $G \cong K_n$ *or* $G \cong \overline{K_n}$*.*

Proposition 5 [25] *Let* G *be a graph of order* n*. Then*

1. $E(G) + E(\overline{G}) \leq n(\sqrt{n} + 1)$
2. $E(G) + E(\overline{G}) \leq (n-1)(\sqrt{n+1} + 1)$ *if* G *is a regular graph.*

Lemma 6 [4] *Let* P*,* Q *be two real symmetric matrices of same order* n *such that* $R = P + Q$*. Then*

$$E(R) \leq E(P) + E(Q),$$

where $E(P) = \sum_{j=1}^{n} |\mu_j|$ *is the energy of* P*, and* μ_j $(j = 1, 2, \ldots, n)$ *are the eigenvalues of* P*.*

3 The exact relation between Seidel energy and energy of a regular graph

In this section, we study the relations between the Seidel energy and the energy of a regular graph. As a consequence, the Seidel equienergetic graphs are constructed by taking regular equienergetic graphs.

Theorem 7 *Let* G *be an* r*-regular graph of order* n*. Then*

$$E_S(G) = |n - 2r - 1| + n - 2r - 1 - 2n^- + 2E(G) + 2 \sum_{\substack{2 \leq j \leq n \\ \theta_j \in (-1/2, 0)}} (2\theta_j + 1).$$

Proof. Let $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_n$ and $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues and the Seidel eigenvalues of a graph G respectively. By definition of the Seidel energy and Theorem 2, we have

$$\begin{aligned}
E_S(G) &= |n - 2r - 1| + \sum_{j=2}^{n} |\lambda_j| = |n - 2r - 1| + \sum_{j=2}^{n} |-1 - 2\theta_j| \\
&= |n - 2r - 1| + \sum_{\substack{2 \leq j \leq n \\ \theta_j \leq -1/2}} (-1 - 2\theta_j) + \sum_{\substack{2 \leq j \leq n \\ \theta_j > -1/2}} (1 + 2\theta_j) \\
&= |n - 2r - 1| - n_\theta[-r, -1/2] + 2 \sum_{\substack{2 \leq j \leq n \\ \theta_j \leq -1/2}} |\theta_j| + n_\theta(-1/2, r) \\
&\quad + 2 \sum_{\substack{2 \leq j \leq n \\ \theta_j \in (-1/2, 0)}} \theta_j + 2 \sum_{\substack{2 \leq j \leq n \\ \theta_j \geq 0}} |\theta_j|. \qquad (1)
\end{aligned}$$

For convenience, the energy of G can be written as

$$E(G) = r + \sum_{\substack{2\le j\le n\\ \theta_j\le -1/2}} |\theta_j| + \sum_{\substack{2\le j\le n\\ \theta_j\ge 0}} |\theta_j| + \sum_{\substack{2\le j\le n\\ \theta_j\in(-1/2,0)}} |\theta_j|.$$

With this, (1) becomes

$$\begin{aligned} E_S(G) &= |n-2r-1| - n_\theta[-r,-1/2] + n_\theta(-1/2,r) + 2\sum_{\substack{2\le j\le n\\ \theta_j\in(-1/2,0)}} \theta_j \\ &\quad + 2\big(E(G) - r - \sum_{\substack{2\le j\le n\\ \theta_j\in(-1/2,0)}} |\theta_j|\big) \\ &= |n-2r-1| - n_\theta[-r,-1/2] + n_\theta(-1/2,r) + 2\sum_{\substack{2\le j\le n\\ \theta_j\in(-1/2,0)}} \theta_j \\ &\quad + 2E(G) - 2r - 2\sum_{\substack{2\le j\le n\\ \theta_j\in(-1/2,0)}} |\theta_j| \\ &= |n-2r-1| + 2E(G) - 2r + 4\sum_{\substack{2\le j\le n\\ \theta_j\in(-1/2,0)}} \theta_j - n_\theta[-r,-1/2] \\ &\quad + n - 1 - n_\theta[-r,-1/2] \\ &= |n-2r-1| + 2E(G) - 2r + 4\sum_{\substack{2\le j\le n\\ \theta_j\in(-1/2,0)}} \theta_j + n - 1 - 2n_\theta[-r,-1/2]. \end{aligned} \tag{2}$$

We have the following

$$n_\theta[-r,-1/2] = n^- - n_\theta(-1/2,0).$$

and

$$\sum_{\substack{2\le j\le n\\ \theta_j\in(-1/2,0)}} (2\theta_j+1) = \sum_{\substack{2\le j\le n\\ \theta_j\in(-1/2,0)}} 2\theta_j + n_\theta(-1/2,0). \tag{3}$$

Now, using (3) in (2)

$$E_S(G) = |n-2r-1| + n - 2r - 1 - 2n^- + 2E(G) + 2\sum_{\substack{2\le j\le n\\ \theta_j\in(-1/2),0)}} (2\theta_j+1)$$

which completes the proof. □

It is easy to see that $\sum_{\substack{2\le j\le n\\ \theta_j\in(-1/2,0)}}(2\theta_j+1)>0$. With this we have the following:

Corollary 8 *Let* G *be an* r*-regular graph of order* n*. Then*

1. $E_S(G) > 2(n-1-2r-n^- + E(G))$ *if* $r \le \frac{(n-1)}{2}$

2. $E_S(G) > 2(E(G)-n^-)$ *if* $r \ge \frac{(n-1)}{2}$.

Corollary 9 *Let* G *be an* r*-regular graph of order* n *and* $\theta_j \notin (-1/2,0)$*. Then*

$$E_S(G) = \begin{cases} 2(n-1-2r-n^- + E(G)) & \text{if } r \le \frac{(n-1)}{2} \\ 2(E(G)-n^-) & \text{if } r \ge \frac{(n-1)}{2}. \end{cases}$$

Proof. If $\theta_j \notin (-1/2,0)$ then $\sum_{\substack{2\le j\le n\\ \theta_j\in(-1/2,0)}}(2\theta_j+1)=0$. Now $r \le \frac{(n-1)}{2}$ implies $n-2r-1\ge 0$ and $r \ge \frac{(n-1)}{2}$ implies $n-2r-1\le 0$. With these, Theorem 7 gives the desired results. □

Remark 10 *In Theorem 3.11 of* [1] *it is proved that* $E_S(G) = 2(n-1-2r-n^- + E(G))$ *if* $\theta_j \notin (-1,0)$ *and* $r \le \frac{n-1}{2}$*. The Corollary 9 gives the same even if* $\theta_j \notin (-1/2,0)$*, which shows that Theorem 3.11 of* [1] *is enriched.*

Corollary 11 *Let* G *be an* r*-regular integral graph of order* n*. Then*

$$E_S(G) = \begin{cases} 2(n-1-2r-n^- + E(G)) & \text{if } r \le \frac{(n-1)}{2} \\ 2(E(G)-n^-) & \text{if } r \ge \frac{(n-1)}{2}. \end{cases}$$

Corollary 12 *Let* G *be an* r*-regular graph of order* n*. Then* $E_S(G) = E(G)$ *if and only if*

$$E(G) = \begin{cases} -2(n-2r-1-n^- + \sum_{\substack{2\le j\le n\\ \theta_j\in(-1/2,0)}}(2\theta_j+1)) & \text{if } r \le \frac{(n-1)}{2} \\ -2(-n^- + \sum_{\substack{2\le j\le n\\ \theta_j\in(-1/2,0)}}(2\theta_j+1)) & \text{if } r \ge \frac{(n-1)}{2}. \end{cases}$$

In a particular case if $\theta_j \notin (-1/2, 0)$, then $E_S(G) = E(G)$ if and only if

$$E(G) = \begin{cases} -2(n - 2r - 1 - n^-) & \text{if } r \leq \frac{(n-1)}{2} \\ 2n^- & \text{if } r \geq \frac{(n-1)}{2}. \end{cases}$$

From the first case of above results and the fact that $E(G) \geq 0$, one can easily observe that if $n - n^- > 2r + 1$ then there is no graph with $E_S(G) = E(G)$. The one of interesting problem in studying the Seidel energy of graphs is to find the Seidel equienergetic graphs. In this direction, we have the following:

Theorem 13 *Let G_1 and G_2 be two equienergetic, r- regular graphs of same order n with no eigenvalue in the interval $(-1/2, 0)$. If G_1 and G_2 both have same number of negative eigenvalues, then G_1 and G_2 are Seidel equienergetic.*

Proof. Proof follows directly from Corollary 9. □

If G is an r-regular graph of order n with $r \geq 3$, then the iterated line graphs $L^k(G)$, $k \geq 2$ have all negative eigenvalues equal to -2 [17]. If G_1 and G_2 are two r-regular graph of order n, where $r \geq 3$ then $L^k(G_1)$, $L^k(G_2)$ have same number of negative eigenvalues and $E(L^k(G_1)) = E(L^k(G_2))$, $k \geq 2$ [20].

Remark 14 *In [19] Ramane et al. studied the Seidel energy of iterated line graphs $L^k(G)$, $k \geq 2$ of a r-regular graph, $r \geq 3$ and constructed a large class of Seidel equienergetic graphs by taking two r-regular graphs of same order. It is noted that the results in [19] become the particular case of Theorem 13.*

4 Seidel energy and energy of a graph

In this section, we study the connection between the Seidel energy and the energy of a graph G in terms of different graph parameters. Also, obtained the inertia relations between the graph eigenvalues and the Seidel eigenvalues of a graph G.

Lemma 15 *Let G be a graph of order n, $n \geq 2$. Then for $j \geq 2$,*

$$2\theta_j + \lambda_{n-j+2} \leq -1 \leq 2\theta_j + \lambda_{n-j+1}. \tag{4}$$

Proof. By definition of Seidel matrix $S = J - I - 2A$ or $2A + S = J - I$. In Lemma 3, by taking $P = 2A$, $Q = S$ and $R = J - I$ we have $\mu_{n-i-k}(J - I) \geq 2\mu_{n-i}(A) + \mu_{n-k}(S)$, now letting $i = n - j, k = j - 2$, we get $\mu_2(J - I) \geq 2\theta_j(A) + \lambda_{n-j+2}(S)$. But $J - I$ has only two different eigenvalues $n - 1$ and $-$

1 which implies $2\theta_j + \lambda_{n-j+2} \leq -1$. Similarly, by letting $s = j-1$ and $t = n-j$ in Lemma 3, we get the right side inequality of (4). □

The inertia of S is the number of positive, negative and 0 eigenvalues of S and denoted by n_S^+, n_S^- and n_S^0 respectively.

Theorem 16 *Let G be a graph of order $n \geq 2$. Then*

(a) $1 \leq n_S^+ + n^+ \leq n+1$

(b) $0 \leq n_S^0 + n^0 \leq n$

(c) $n-1 \leq n_S^- + n^-$.

Proof. The $n_S^+ = 1$ if and only if $G \cong K_{n_1,n_2}(n_1+n_2=n)$ or $\overline{K_n}$ [5] which implies $\lambda_2 \geq 0$ for remaining graphs. This shows that lower bound in (a) is clear. The lower bound in (b) follows from the fact that many graphs are not Seidel integral. Let us rewrite the eigenvalues of G as

$$\theta_1 \geq \cdots \geq \theta_{n^+} > 0 = \theta_{n^++1} = \cdots = \theta_{n^++n^0} = 0 > \theta_{n^++n^0+1} \geq \cdots$$
$$\cdots \geq \theta_{n^++n^0+j} > -\frac{1}{2} \geq \theta_{n^++n^0+j+1} \geq \cdots \geq \theta_n.$$

From the left side inequality of (4), we have $2\theta_j + \lambda_{n-j+2} \leq -1$ for $j \geq 2$. And if $j \geq 2$ and $\theta_j > -\frac{1}{2}$ then $\lambda_{n-j+2} \leq -1 - 2\theta_j < 0$.
Therefore,

$$n_S^- \geq n^+ + n^0 + j - 1 \tag{5}$$

which implies

$$n_S^- + n^- \geq n^+ + n^0 + j - 1 + n^- = n + j - 1 \geq n - 1.$$

Now by using (5),

$$n_S^+ + n^+ = n - n_S^- - n_S^0 + n^+ \leq n - (n^+ + n^0 + j - 1) - n_S^0 + n^+ \leq n + 1.$$

Next, by using above bounds

$$n_S^0 + n^0 = 2n - (n_S^+ + n^+) - (n_S^- + n^-) \leq 2n - 1 - (n-1) = n.$$

□

In the following result we give necessary and sufficient condition to hold the equality in left side of (4).

Lemma 17 *Let* G *be a graph of order* n*. Then* $2\theta_j + \lambda_{n-j+2} = -1$ *for all* $j \in \{2, 3, \ldots, n\}$ *if and only if* G *is an* r*-regular graph with largest Seidel eigenvalue* $n - 2r - 1$*.*

Proof. If G is an r-regular graph. Then by Theorem 2 it is clear that $2\theta_j + \lambda_{n-j+2} = -1$ for all $j \in \{2, 3, \ldots, n\}$. Conversely, assume that $2\theta_j + \lambda_{n-j+2} = -1$ for all $j \in \{2, 3, \ldots, n\}$. Let the number of edges in G and its complement $\overline{G}$ be m and $\overline{m}$ respectively. By Rayleigh quotient with all one's vector $\mathbf{1}$, we have

$$\lambda_1 = \max_{x \neq 0}\{\frac{x^T S x}{x^T x}\} \geq \frac{\mathbf{1^T}(\overline{A} - A)\mathbf{1}}{\mathbf{1^T 1}} \geq \frac{2\overline{m} - 2m}{n}.$$

From the fact that $\sum_{j=1}^{n} \theta_j = 0$ and $\sum_{j=1}^{n} \lambda_j = 0$, we have

$$\sum_{j=1}^{n} (2\theta_j + \lambda_j) = 0 \text{ which implies } 2\theta_1 + \lambda_1 = n - 1.$$

On the other hand, from the fact $\theta_1 \geq \frac{2m}{n}$, we have

$$2\theta_1 + \lambda_1 \geq \frac{4m}{n} + \frac{2\overline{m} - 2m}{n} = n - 1.$$

Now we arrive at $\theta_1 = \frac{2m}{n}$ and $\lambda_1 = \frac{2\overline{m} - 2m}{n}$, which shows that G is an r-regular graph with $r = \frac{2m}{n}$ and $\lambda_1 = n - 2r - 1$.
This completes the proof. □

Proposition 18 *Let* G *be a graph of order* n*. Then*

$$|E_S(G) - 2E(G)| \leq 2n - 2.$$

Equality holds if and only if G *is a complete graph.*

Proof. By definition of Seidel matrix, $S = J - I - 2A$. Now using Lemma 6, we have, $E(S) \leq E(J - I) + 2E(A)$ and $2E(A) \leq E(J - I) + E(S)$ which implies $|E_S(G) - 2E(G)| \leq 2n - 2$. □

Theorem 19 *Let* G *be a graph of order* n*. Then*

(a) $2E(G) - E_S(G) \leq 4\theta_1$

(b) $E_S(G) - 2E(G) \leq 2\lambda_1$.

Proof. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ and $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_n$ are the Seidel eigenvalues and the eigenvalues of a graph G respectively. From the definition of the Seidel energy and the energy of a graph G, we have

$$E_S(G) = \lambda_1 + \sum_{j=2}^{n} |\lambda_j|$$

$$2E(G) = 2\theta_1 + \sum_{j=2}^{n} |2\theta_{n-j+2}|.$$

By combining the above two equalities we get

$$\begin{aligned} 2E(G) - 2\theta_1 - E_S(G) + \lambda_1 &= \sum_{j=2}^{n} (|2\theta_{n-j+2}| - |\lambda_j|) \\ &\leq \sum_{j=2}^{n} |2\theta_{n-j+2} + \lambda_j|. \end{aligned}$$

Now from the left side inequality of (4), we have $2\theta_j + \lambda_{n-j+2} \leq -1$ for $j \geq 2$. Therefore

$$\sum_{j=2}^{n} |2\theta_{n-j+2} + \lambda_j| = -\sum_{j=2}^{n} 2\theta_{n-j+2} + \lambda_j = 2\theta_1 + \lambda_1.$$

With this fact we arrive at $2E(G) - E_S(G) \leq 4\theta_1$.
Similarly one can prove (b) easily. □

Remark 20 *The case* (b) *of Theorem 19 with the fact* $\lambda_1 \leq n - 1$ *gives*

$$E_S(G) - 2E(G) \leq 2\lambda_1 \leq 2n - 2.$$

And Proposition 18 gives that $E_S(G) - 2E(G) \leq 2n - 2$ *which implies the result* (b) *of Theorem 19 is better than this inequality.*

Theorem 21 *Let* G *be a graph of order* n *with Seidel eigenvalues* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. *Then*

$$2E(G) - E_S(G) \leq 4\theta_1 - 2n + 2n^- + 2. \tag{6}$$

Equality holds if and only if G *is a graph with* $2\theta_j + \lambda_{n-j+2} = -1$ *for all* $j \in \{2, 3, \ldots, z\}$ *and* $n^- + n_S^- = n - 1$, *where* $z = n^+ + n^0$.

Proof. Let $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_n$ be the eigenvalues of G. From the definition of energy of a graph G, we have

$$\begin{aligned} E(G) &= \sum_{j=1}^{n} |\theta_j| = 2\sum_{j=1}^{z} \theta_j = 2\theta_1 + 2\sum_{j=2}^{z} \theta_j \\ &\leq 2\theta_1 + 2\sum_{j=2}^{z} \frac{1}{2}(-1 - \lambda_{n-j+2}) \quad \text{by left side of (4)} \qquad (7) \\ &= 2\theta_1 - z + 1 - \sum_{j=1}^{z-1} \lambda_{n-j+1} \\ &\leq 2\theta_1 - z + 1 + \max_{1\leq i\leq n} \sum_{j=1}^{i} -\lambda_{n-i+1} \qquad (8) \\ &= 2\theta_1 - n + n^- + 1 + \frac{1}{2} E_S(G) \\ 2E(G) - E_S(G) &\leq 4\theta_1 - 2n + 2n^- + 2. \end{aligned}$$

For equality we have the following. Let G be a complete graph of order n, then $\theta_1 = n - 1$ and $n^- = n - 1$. Hence equality holds in (6). Suppose G is not a complete graph then $\theta_2 \geq 0$, which gives $z = n^+ + n^0 = n - n^- \geq 2$.
Now, to have equality in (6) the inequalities (7) and (8) must be equalities. Equality in (7) holds if and only if $2\theta_j + \lambda_{n-j+2} = -1$ for all $j \in \{2, 3, \ldots, z\}$. From the energy of a graph and equality in (8) holds if and only if

$$n_S^- \leq z - 1 \leq n_S^- + n_S^0.$$

Since $z = n - n^-$ we get $n_S^- + n^- \leq n - 1$, and $n_S^- + n^- \geq n - 1 - n_S^0$. Now, from (c) of the Theorem 16, the above right side inequality is obvious. Again using (c) of the Theorem 16 with left side inequality of above, we have $n^- + n_S^- = n - 1$. This completes the proof. □

Corollary 22 *Let G be a graph of order n. Then*

$$2E(G) - E_S(G) \leq 4\Delta - 2n + 2n^- + 2.$$

Equality holds if and only if G is regular graph with $n^- + n_S^- = n - 1$.

Proof. Let Δ be the maximum degree of G. It is well-known fact that $\theta_1 \leq \Delta$. And $\theta_1 = \Delta$ holds if and only if G is a regular. Using these in Theorem 21, we arrive at required results. □

Remark 23 *From the fact that* $n^- \leq n-1$, *Theorem 21 gives*

$$2E(G) - E_S(G) \leq 4\theta_1 - 2n + 2n^- + 2 \leq 4\theta_1.$$

And the case (a) *of Theorem 19 shows that* $2E(G) - E_S(G) \leq 4\theta_1$ *which implies the result in Theorem 21 is better than this inequality.*

Corollary 24 *Let* G *be a graph of order* n. *Then*

$$2E(G) - E_S(G) \leq 2(n - 1 + n^-) \leq 2(2n - 1 - \alpha). \tag{9}$$

Proof. Let α be the independence number of G. Using the fact that $\theta_1 \leq n-1$ in Theorem 21 the left side inequality in (9) is clear. Now using the well known inequality $n^- \leq n - \alpha$, we get the right side inequality in (9). □

Theorem 25 *Let* G *be a graph of order* n. *Then*

(a) $2(n^+ + n^0 - 1) \leq E_S(G) \leq n(\sqrt{n} + 1)$

(b) $2(n^+ + n^0 - 1) \leq E_S(G) \leq (n-1)(\sqrt{n+1} + 1)$ *if* G *is a regular graph.*

Proof. Left side inequality in (a) and (b) follows by using the fact $E(G) \geq 2\theta_1$ in (6). Now by using the right side inequality of Theorem 4 and (1) of Proposition 5, we have the right side inequality of (a). Now by using the right side inequality of Theorem 4 and (2) of Proposition 5, we have the right side inequality of (b). □

References

[1] S. Akbari, J. Askari, K. C. Das, Some properties of eigenvalues of the Seidel matrix, *Linear Multilinear Algebra* (2020) 1–12. DOI:10.1080/03081087.2020.1790481. ⇒48, 51

[2] S. Akbari, M. Einollahzadeh, M. M. Karkhaneei, M. A. Nematollahi, Proof of a conjecture on the Seidel energy of graphs, *European J. Combin.* **86** (2020) 103078. ⇒48

[3] A. E. Brouwer, W. H. Haemers, *Spectra of Graphs*, Springer, Berlin, 2012. ⇒48

[4] K. Fan, Maximum properties and inequalities for the eigenvalues of completely continuous operators, *Proc. Natl. Acad. Sci USA.* **37** (1951) 760–766. ⇒49

[5] M. Ghorbani, M. Hakimi-Nezhaad, B. Zhou, Graphs with at most four Seidel eigenvalues, *Kragujevac J. Math.* **47** (2023) 173 –186. ⇒53

[6] I. Gutman, O. E. Polansky, *Mathematical Concepts in Organic Chemistry*, Springer, 2012. ⇒47
[7] I. Gutman, The energy of a graph, *Ber. Math. Stat. Sekt. Forschungsz. Graz.* **103** (1978) 1–22. ⇒47
[8] W. H. Haemers, Seidel switching and graph energy, *MATCH Commun. Math. Comput. Chem.* **68**, 3 (2012) 653-659. ⇒47
[9] F. Harary, *Graph Theory*, Narosa Publishing House, New Delhi, 1998. ⇒48
[10] R. A. Horn, C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 2012. ⇒48
[11] G. Indulal, Distance equienergetic graphs, International Conference on Graph Connections (ICGC)-2020, Bishop Chulaparambil Memorial College and Mahatma Gandhi University. https://icgc2020.wordpress.com/invitedlectures (2020). Accessed 8 August 2020. ⇒47
[12] X. Li, Y. Shi, I. Gutman, *Graph Energy*, Springer, New York, 2012. ⇒47
[13] H. S. Ramane, K. Ashoka, D. Patil, B. Parvathalu, On the Seidel Laplacian and Seidel signless Laplacian polynomials of graphs, *Kyungpook Math. J.* **61** (2021) 155–168. ⇒48
[14] H. S. Ramane, K. Ashoka, B. Parvathalu, D. Patil, On A-energy and S-energy of certain class of graphs, *Acta Univ. Sapientiae Informatica* **13**, 2 (2021) 195–219. DOI:10.2478/ausi-2021-0009 ⇒48
[15] H. S. Ramane, B. Parvathalu, K. Ashoka, Seidel energy of k-fold and strong k-fold graphs, *Commun. Combin., Cryptogr. & Computer Sci.* **1**(2021) 152–159. DOI:10.5666/KMJ.0000.00.0.000. ⇒48
[16] H. S. Ramane, B. Parvathalu, K. Ashoka, S. Pirzada, On families of graphs which are both adjacency equienergetic and distance equienergetic, *Indian J Pure Appl Math*. DOI:10.1007/s13226-022-00355-1. ⇒47
[17] H. S. Ramane, B. Parvathalu, D. Patil, K. Ashoka, Iterated line graphs with only negative eigenvalues -2, their complements and energy, *Preprint* (2022). DOI:10.48550/arXiv.2205.02276. ⇒52
[18] H. S. Ramane, M. M. Gundloor, S. M. Hosamani, Seidel equienergetic graphs, *Bull. Math. Sci. Appl.* **16** (2016) 62–69. ⇒48
[19] H. S. Ramane, I. Gutman, M. M. Gundloor, Seidel energy of iterated line graphs of regular graphs, *Kragujevac J. Math.*, **39**, 1 (2015), 7–12. ⇒52
[20] H. S. Ramane, H. B. Walikar, S. B. Rao, B. D. Acharya, P. R. Hampiholi, S. R. Jog, I. Gutman, Spectra and energies of iterated line graphs of regular graphs, *Appl. Math. Lett.* **18** (2005) 679–682. ⇒52
[21] J. J. Seidel, Strongly regular graphs with $(-1, 1, 0)$ adjacency matrix having eigenvalue 3, *Linear Algebra Appl.* **1** (1968) 281–298. ⇒47
[22] G. X. Tian, Y. Li, S. Y. Cui, The change of Seidel energy of tripartite Turán graph due to edge deletion, *Linear Multilinear Algebra* (2021) 1–18. DOI:10.1080/03081087.2021.1888858. ⇒48
[23] S. K. Vaidya, K. M. Popat, Some new results on Seidel equienergetic graphs, *Kyungpook Math. J.* **59**, 2 (2019) 335—340. ⇒48

[24] J. H. van Lint, J. J. Seidel, Equilateral point sets in elliptic geometry, *Indag Math.* **28** (1966) 335–348. ⇒47

[25] B. Zhou, I. Gutman, Nordhaus-Gaddum type relations for the energy and Laplacian energy of graphs, *Bull. Cl. Sci. Math. Nat. Sci. Math.* (2007) 1–11. DOI: 10.1071/ASEG2007ab174. ⇒49

Received: May 3, 2023 • *Revised: May 23, 2023*

Acta Univ. Sapientiae Informatica **15,** 1 (2023) 60–80

DOI: 10.2478/ausi-2023-0006

Explainable patch-level histopathology tissue type detection with bag-of-local-features models and data augmentation

Gergő GALIGER
Babeş–Bolyai University
Cluj-Napoca, Romania
email: galigergergo@yahoo.com

Zalán BODÓ
Babeş–Bolyai University
Cluj-Napoca, Romania
email: zalan.bodo@ubbcluj.ro

Abstract. Automatic detection of tissue types on whole-slide images (WSI) is an important task in computational histopathology that can be solved with convolutional neural networks (CNN) with high accuracy. However, the black-box nature of CNNs rightfully raises concerns about using them for this task. In this paper, we reformulate the task of tissue type detection to multiple binary classification problems to simplify the justification of model decisions. We propose an adapted Bag-of-local-Features interpretable CNN for solving this problem, which we train on eight newly introduced binary tissue classification datasets. The performance of the model is evaluated simultaneously with its decision-making process using logit heatmaps. Our model achieves better performance than its non-interpretable counterparts, while also being able to provide human-readable justification for decisions. Furthermore, the problem of data scarcity in computational histopathology is accounted for by using data augmentation techniques to improve both the performance and even the validity of model decisions. The source code and binary datasets can be accessed at: https://github.com/galigergergo/BolFTissueDetect.

Key words and phrases: computational pathology, tissue type detection, explainable artificial intelligence, convolutional neural networks, data augmentation

1 Introduction

The accurate interpretation and analysis of medical images can aid in the diagnosis of various diseases and conditions [9], provide information about the progression of diseases [24], and assist in treatment planning [25]. In the field of histopathological image processing, digital images are obtained as a result of the invasive technique of removing and scanning specific tissue samples from a patient's body through some form of biopsy [8]. The scanned whole-slide images (WSI) represent high-resolution images of gigapixel order, which results in a very high computational cost of analyzing them with different algorithms [16]. To solve the problems caused by WSI sizes, single whole-slide images can be broken down into several smaller patches, and then processed in adequately sized batches depending on available resources.

The introduction of WSIs made it possible to develop different computer-aided diagnosis (CAD) and prognosis systems for automating various tasks in the field of medical image processing [15]. The task in the field of WSI processing examined in this paper is cancer detection, more specifically the detection of different tissue types in WSIs of tumor microenvironments (TME). TME has a significant impact on tumor initiation and progression [12] and also affects the prognosis and response to therapy of cancer patients [1].

Automating different tasks of trained human pathologists using CAD systems poses a variety of different challenges by itself. For instance, machine learning-based methods may be vulnerable to only changing a single pixel of an image [19]. In case of WSIs different artifacts (air bubbles, dirt) may contaminate tissue samples, and different forms of deterioration (tears, cracks, color variations) of the samples are also common [15], which could all bias machine-learning models. Furthermore, creating a large, high-quality annotated medical dataset, which is essential for all tasks involving supervised learning, is also considered a difficult task. Correct annotation of WSIs is a time-consuming, costly process that requires the involvement of a large number of trained professionals [2]. Moreover, privacy concerns may arise when working with specific medical conditions [20]. Another major challenge of using CAD systems in the medical sector arises from the black-box nature of the current state-of-the-art approaches [22]. Although deep learning methods have shown promising performances in medical tasks, the inability to explain decisions has raised concern among medical experts [21].

Due to task-related complexities, histopathological segmentation methods might rely on additional, a-priori information besides the input images. As a related example, in [11], the proposed state-of-the-art TME tissue segmenta-

tion method integrates the usage of classification labels of the input images to achieve good performance. However, no method of identifying these classification labels was introduced in this paper, the authors implying the task to be carried out by human radiologists. Thus, the proposed method represents a semi-automated approach, which heavily relies on human pathologists identifying different tissue types in WSIs.

The main focus of this paper is to fully automate this process by introducing a way of identifying the TME tissue types used as inputs for the segmentation method proposed in [11]. As this approach intends to replace human experts, the reliability of its decision-making process is of utmost importance. Therefore, we obtain highly interpretable problems by reformulating the task of TME tissue type detection as multiple binary classification problems. We refer to the interpretability of model decisions as the correspondence of the model activations and the ground truth. For solving the task of binary tissue classification, we propose an adapted version of the explainable BagNet architecture [5], which demonstrates matching performance compared to human experts. Moreover, the usage of the BagNet architecture also provides a solution for the concerns regarding the black-box nature of models in the medical field by being able to justify decisions taken. The proposed models show a decision-making process that aligns with the expected histopathological reasoning of human radiologists. In addition, we solve the problem of medical data scarcity with data augmentation techniques, the effects of which on both the performance and validity of model decisions are further analyzed in detail.

Overall, we present four main contributions in this paper: (i) The task of TME tissue type detection is reformulated to multiple binary classification tasks for the sake of clear, human-readable justification of model decision-making. (ii) The BagNet Bag-of-local-Features explainable CNN architecture is adapted for this task and the validity of its decisions is evaluated using logit heatmaps. (iii) Eight binary datasets are created for this task from two publicly available ones and the effects of expanding them using data augmentation techniques are also reviewed. (iv) A detailed quantitative and qualitative evaluation of the adapted method is performed on the newly created datasets.

The rest of this paper is structured as follows: Section 2 describes the specific TME binary classification task formulation and necessary adaptations in both model architecture and dataset structure. In Section 3, we present the experiments used to benchmark the adapted BagNet model on the binary classification task, along with an in-depth analysis of obtained results. We conclude our findings in Section 4, where we also provide further research directions regarding this topic.

2 Methodology

2.1 Problem formulation

For the task of identifying patch-level labels, such as in the case of the training samples from the two presented datasets, individual patches have to be assigned to four different classes, thus resulting in a multi-class multi-label classification task. Successfully solving this problem, while also providing reliable explanations for decisions is highly difficult in the case of such a complex classification task. Therefore, the task should be reformulated to solve one binary classification task for each of the four classes, i.e. detecting whether or not one type of tissue (class) is present in a certain patch. This reformulation yields multiple significantly simpler detection problems, which in turn simplify the interpretation process of explainable models used for solving these.

Our choice for solving these binary classification problems was the BagNet interpretable CNN architecture, which showed promising results in a variety of different tasks [5, 17, 18] and is inherently explainable with pixel-level activation heatmaps. As we are using weakly-annotated datasets with fully segmented validation and testing data, the exact location of tissue types in these cases is precisely known. This implies the natural application of heatmaps for decision interpretation since pixel activation validity can be verified using segmentation data.

BagNets were originally proposed for single-label multi-class classification tasks on the ImageNet dataset [7], which makes adapting it to binary classification problems a necessary procedure. This was done by changing the output size of the linear classifier following the average pooling operator to 1 and replacing the softmax operator with sigmoid to infer the image-level class evidence. By introducing the sigmoid operator, the independence of different classes is assumed, which is inherent in the case of the two classes of this binary classification problem: a certain tissue type is present in a patch versus it is not present.

2.2 Datasets

The scarcity of suitable quality datasets for medical image processing applications applies to WSIs of tumor microenvironments, which requires radiology expertise to be correctly annotated [2], and the publication of such datasets also raises concerns regarding patient privacy [20]. The datasets used in this paper originate from the previously mentioned paper [11], where two different

Binary dataset	Training	Testing	Validation
LUAD – LYM	1400	200	100
LUAD – NEC	6000	120	40
LUAD – TE	8000	80	60
LUAD – TAS	8000	140	40
BCSS – NEC	6000	800	800
BCSS – LYM	12 000	4000	2000
BCSS – STR	16 000	4000	1600
BCSS – TUM	18 000	4000	3000

Table 1: Size distribution of the 8 binary datasets between training, testing, and validation subsets.

weakly-annotated datasets of TMEs were created and published for further research:

LUAD-HistoSeg (LUAD): This dataset was specifically created for training the proposed segmentation model and included patches with four tissue categories: tumor epithelial (TE), tumor-associated stroma (TAS), necrosis (NEC) and lymphocyte (LYM).

BCSS-WSSS (BCSS): This dataset was adapted from a previously published fully-supervised dataset [2] and included four classes: tumor (TUM), stroma (STR), lymphocytic infiltrate (LYM), and necrosis (NEC).

In a similar manner to the used model architecture, the two original datasets would have been adequate for a different type of problem, a multi-class multi-label classification task. This resulted in both of the datasets having to be adapted to the newly formulated binary classification problems. The adaptation consisted in creating a binary classification dataset for each of the classes from the original datasets. This was done by first counting the positive and negative examples for a given class in a dataset, i.e. the number of samples that contained a given tissue type and the number of samples that did not. To obtain an optimal balance between the two classes, the binary datasets were constructed with the smaller one of the two binary classes (positive or negative) and a randomly sampled version of the larger binary class. Applying this method to all four classes of both datasets resulted in 8 (4×2) binary datasets of varying sizes. Table 1 shows the size distribution of the 8 binary datasets.

Transformation	Probability
Horizontal skew	1.0
Low-angle rotation	1.0
90-degree-multiple rotation	0.75
Horizontal flip	0.5
Vertical flip	0.5

Table 2: Probabilities of applying different data augmentation transformations when expanding binary datasets.

2.3 Data augmentation

The original LUAD and BCSS datasets are relatively small for training DNNs in a supervised manner, even for the task of binary classification. The problem of data scarcity is further accentuated by the fact that the actual datasets used for binary classification are even smaller than the original ones. This happens because these represent subsets of the original datasets obtained by their adaptation to the binary classification task, four binary datasets having been created from a single original dataset. To alleviate the issue of small datasets, we used data augmentation methods to expand all eight binary datasets to 10 times their original size.

To simulate common, practically occurring changes in WSIs such as different rotation and skew angles of slides during the scanning procedure, only related augmentation transformations were used. The augmentation techniques used for the expansion of the binary datasets were limited to morphological transformations, which included low-magnitude skew operations, random-angle rotations by a maximum of 3 degrees, rotations with 90-degree multiples, and flip operations. All of the listed transformations were applied with certain probabilities, as shown in Table 2. These values were chosen based on the following reasoning: skew and low-angle rotation are always applied to introduce some level of morphological diversity in every single new image, while the remaining three transformations further differentiate images, the probability for rotation being higher since flip operations are twice as many in number.

2.4 Implementation details

All the convolutional neural networks analyzed in our experiments were implemented in PyTorch and we used the Augmentor[1] Python package for aug-

[1]Augmentor Python Package (https://augmentor.readthedocs.io/en/stable)

menting the binary datasets. We trained the models using an NVIDIA Titan Xp GPU and utilized ResNet-50 as the classification backbone for the BagNet architecture. The resolution of the input patches was 224 × 224 and the batch size was set to 29, the maximum value we could obtain before reaching GPU memory limitations. The network weights were optimized for binary cross-entropy loss using stochastic gradient descent (SGD) with a weight decay of 0.0001 and a momentum of 0.9. The learning rate was set to 0.01 and divided by 10 every 30 training epochs. These hyperparameter settings were inspired by the ImageNet training example from the original PyTorch GitHub repository[2].

3 Experiments and results

3.1 Training process

3.1.1 Receptive field size

The main advantage of using the BagNet architecture lies in the simplicity of explaining its decision-making process, which is a consequence of its separation of receptive fields in inferring class evidence. From the originally proposed BagNet–q models, with $q \in \{9, 17, 33\}$ receptive field sizes, BagNet–17 showed the best classification performance on the ImageNet dataset [5]. However, the transferability of ImageNet results to WSI processing is questionable due to the inherent differences between natural images and tissue scans. As a result, we benchmarked all three BagNet models for binary tissue classification on the LUAD – LYM dataset to find the most suitable receptive field size for this application. The results of this experiment are shown in Table 3.

In contrast to the natural images from ImageNet, where the BagNet–17 model showed clearly superior classification performance, the results are less clear in this case. The smaller receptive field size of 9 × 9 pixels leads to the highest values in some of the calculated performance measures, although these are not significantly better than the ones obtained with 17 × 17 receptive fields, with below 2% differences in average validation precision, recall, and specificity. However, there are significant differences in performance in every other case where the values obtained by BagNet–17 are highest, mostly exceeding 5%, and even 10% in some cases. As a result of these findings, we use the BagNet–17 model in the following experiments, which we will also refer to as BagNet.

[2]PyTorch GitHub Repository (https://github.com/pytorch/examples/tree/main/imagenet)

Model	Acc.	Prec.	Recall	Spec.
Without data aug.				
BagNet–9	0.93	0.875	**0.9667**	**0.9459**
BagNet–17	**0.94**	**0.925**	0.95	0.925
BagNet–33	0.88	0.8	0.9333	0.8889
With data aug.				
BagNet–9	0.908	**0.9225**	0.8983	0.8581
BagNet–17	**0.954**	0.9175	**0.9783**	**0.966**
BagNet–33	0.873	0.7675	0.9433	0.9003

Table 3: Results of training BagNet models with different receptive field sizes. The values shown in this table are the average validation accuracy, precision, recall, and specificity scores obtained after training the models on the LUAD – LYM dataset with and without using data augmentation.

3.1.2 Model backbone

Neural networks show a tendency to benefit from increasing depth in various image processing tasks [10]. In [5], the BagNet network was proposed with the ResNet-50 architecture as a backbone, which is considered a highly layered architecture by current standards [13]. In order to test the necessity of such a deep backbone architecture for the binary classification task of TME tissue types, we trained the BagNet model on the LUAD – LYM dataset using two different backbones: the previously mentioned ResNet-50, and a significantly smaller CNN, which consisted of two convolutional layers of 32 and 64 neurons respectively, followed by a ReLU activation layer.

To present the findings of this experiment, the average validation accuracy values of the two models are illustrated on the left side of Figure 1. Using the small CNN as a backbone, the accuracy values converge in around 30 epochs, not improving considerably on initial values. This could be caused by the small network's inability to sufficiently generalize the complex image processing problem. In contrast, the significantly deeper ResNet-50 backbone leads to later conversion but shows substantially improved classification performance. Therefore, we use the ResNet-50 as the backbone for our BagNet models in the following experiments.

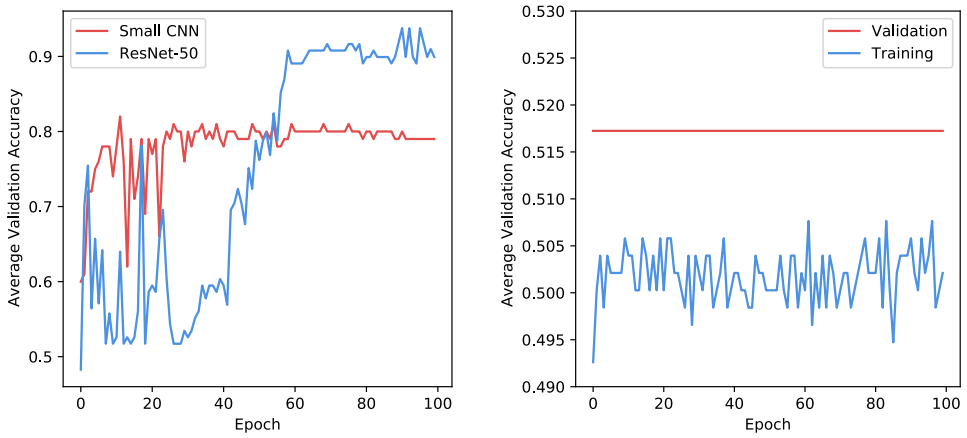


Figure 1: Results of benchmarking small CNN and ResNet-50 backbones for BagNet for binary classification trained on the LUAD – LYM dataset.

3.1.3 Transfer learning

Likewise to many other architectures, leveraging transfer learning, i.e. initializing BagNet weights with weights originally trained on the ImageNet dataset leads to significant performance increases in many different tasks [4, 17]. To verify the validity of the premise for the task of TME tissue binary classification, we evaluated three different ways of weight initialization for the adapted BagNet model:

Default initialization: No pre-trained weights were loaded in this case, initialization was done by the default PyTorch weight initialization process.

Loading ImageNet pre-trained weights: In this case, weights were loaded from the original pre-trained BagNet model published with the original paper [5]. As with the network architecture, adaptations had to be carried out in the case of the pre-trained weights as well, since the last linear transformation layer of the architectures did not match. The adaptation consisted in averaging the weights corresponding to the 2048 ImageNet classes in order to obtain a single weight tensor.

Loading LUAD – NEC pre-trained weights: We managed to obtain promising results by training the BagNet model with default initialization on the LUAD – NEC binary dataset. The weights of this pre-trained model were also used in this benchmark for initialization for training on binary datasets different from LUAD – NEC.

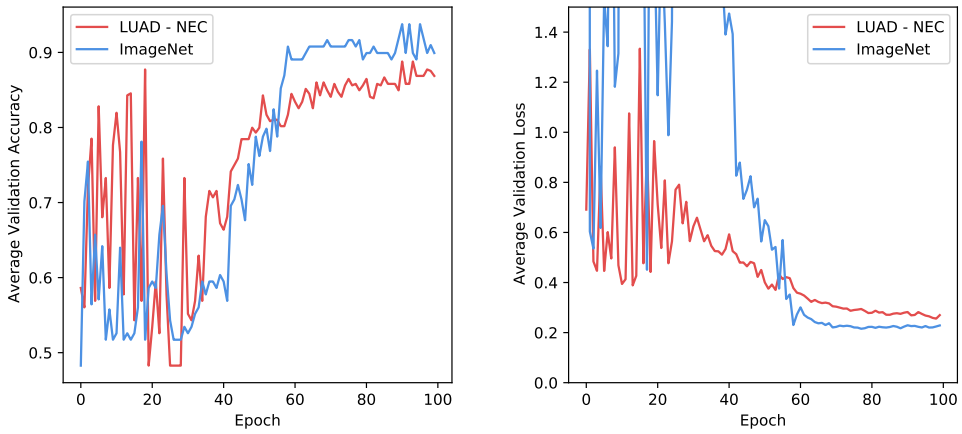


Figure 2: Results of training BagNet with ResNet-50 backbone on the LUAD – LYM dataset with weight initialization using pre-trained models on ImageNet and LUAD – NEC datasets.

These weight initialization approaches were benchmarked by using them for training on the LUAD – LYM binary dataset. The right side of Figure 1 shows training and validation results obtained by training the BagNet model with no weight initialization. The model is unable to escape from a local optimum during 100 epochs of optimization, the training values are slightly oscillating, while the validation values stay constant. This phenomenon might be caused by unlucky weight initialization, however, three separate runs of this experiment showed similar behavior.

The results of training the BagNet model with the two different pre-trained sets of weights are shown in Figure 2. Both approaches show similar behavior in terms of accuracy and loss, both converging after approximately 60 epochs, however, ImageNet weight initialization leads to slightly better classification performance. This demonstrates the generalizability of deep networks such as ResNet-50, which was able to successfully transfer knowledge learned on the ImageNet dataset to the completely different task of WSI binary classification. The underperformance of LUAD – NEC weights might be caused by the significantly lower training time of this model (approximately 400 epochs) compared to the original BagNet model, which was trained for over 5000 epochs on the ImageNet dataset [5].

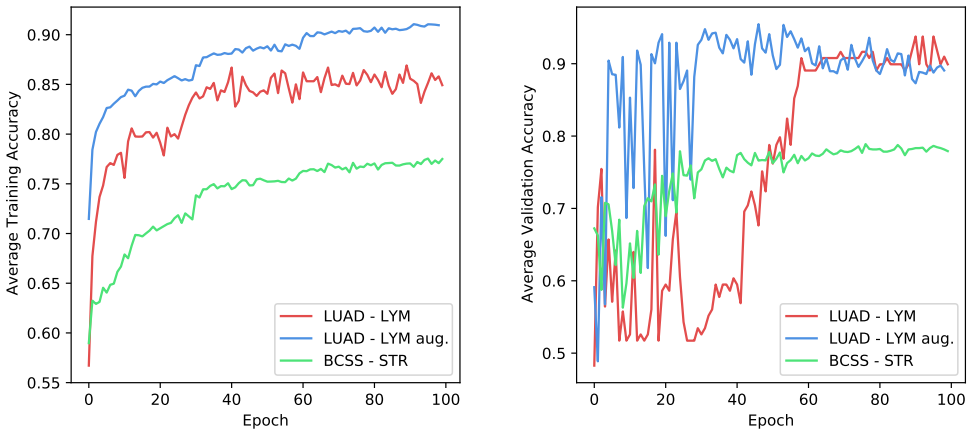


Figure 3: Results of training BagNet with ResNet-50 backbone, ImageNet weight initialization on datasets of different sizes: LUAD – LYM with 1400 training samples and 100 validation samples, augmented LUAD – LYM with 14 000 training samples, and 1000 validation samples, BCSS – STR with 16 000 training samples, and 1600 validation samples.

3.1.4 Dataset properties

The training speed and convergence rate, as well as the ability of neural networks to learn generalization of task-related information, are all dependent on the quality and size of the dataset used for training them. As the eight binary datasets created for the task of binary TME tissue classification are all different in size, we evaluate the effect of dataset size on the training and validation process of the adapted BagNet model. This is done by comparing training and validation accuracy values of three different binary datasets over 100 training epochs.

Figure 3 shows the results of the dataset size examination carried out on LUAD – LYM, augmented LUAD – LYM and BCSS – STR, with 1400, 14 000 and 16 000 training samples respectively. By analyzing the results obtained from this experiment, we managed to draw two different conclusions, one regarding the size of two completely different datasets and one regarding the size difference obtained by using data augmentation.

On the one hand, by comparing the training process on the smallest binary dataset, LUAD – LYM, and one of the largest non-augmented datasets, BCSS – STR, we can observe that a larger dataset leads to a slower and more gradual convergence both in case of training and validation. The smaller dataset ap-

pears to show significantly better validation performance, however, this might be heavily influenced by the significantly smaller size of the validation dataset, 100 samples versus 1600 samples for BCSS – STR. Both validation and accuracy metrics could also be skewed by the fact that the original BCSS dataset was significantly less detailed since it was adapted from a more general dataset [11].

On the other hand, the effects of data augmentation on the training process could also be evaluated by comparing the results on the non-augmented and augmented versions of the LUAD – LYM dataset. Augmenting this small-sized dataset leads to faster convergence and better classification performance including unseen data. However, the model appears to be overfitting the training data after approximately 50 epochs, where the validation accuracy starts showing a descending trend, while the training accuracy is still increasing. This is most likely caused by the low diversity of the augmented dataset, which is a result of applying morphological transformations to the LUAD – LYM binary dataset.

3.2 Quantitative evaluation

In order to benchmark the adapted BagNet architecture for the task of binary classification of TME tissues, the model was first trained on datasets of different sizes with and without using data augmentation, followed by an evaluation on previously unseen data. The results of this experiment are shown in Table 4, along with the sizes of datasets the models were trained and validated on.

By analyzing the performance measure scores obtained in this experiment, we can draw some conclusions about the usability of the BagNet architecture for this specific binary classification task. All models trained on the binary datasets derived from the original LUAD dataset showed promising results with accuracy values starting at 0.94 on previously unseen data, however, the relatively small size of the validation datasets has to be taken into account in this case as well. In the case of the BCSS – STR dataset, which contains a slightly larger validation subset, the BagNet model obtained significantly lower scores than the models trained on the LUAD datasets. This might be caused by the BCSS dataset being less detailed, a topic that is further discussed in Section 3.3.

By comparing the results obtained on augmented datasets with their non-augmented counterparts in the first four rows of Table 4, an increase in accuracy values can be observed in both cases, which indicates that the usage of data augmentation leads to improvements in classification performance.

Dataset	Acc.	Prec.	Recall	Spec.	Train s.	Val. s.
LUAD –						
LYM	0.94	0.925	0.925	0.95	1400	100
LYM aug.	0.954	0.966	0.9175	0.9783	14 000	1000
NEC	0.975	0.9091	1.0	0.9667	6000	40
NEC aug.	0.985	0.9434	1.0	0.98	60 000	400
TE aug.	0.9467	1.0	0.9135	1.0	80 000	600
BCSS –						
STR	0.7888	0.8368	0.7732	0.8085	16 000	1600

Table 4: Results of training BagNet with ResNet-50 backbone with ImageNet weight initialization on binary datasets of different sizes with and without data augmentation. The values shown in this table are the average validation accuracy, precision, recall, and specificity, along with training and validation subset sizes for different binary datasets.

Moreover, the results presented in this table demonstrate comparable performances to radiology experts, in most cases even exceeding the capabilities of human pathologists [14], which shows real-world applicability. However, the robustness of the model on WSI-specific artifacts, color variations, or slide deteriorations mentioned in Section 1 has not been tested, which might limit model usability in real-world environments.

The BagNet model has also been compared to the non-interpretable convolutional networks VGG19 and the backbone used for all BagNet models in this paper, ResNet50. We trained and validated these models on the LUAD – NEC binary dataset with and without transfer learning, and summarized the obtained results in Table 5 using four different performance metrics. Although the added interpretability of the BagNet architecture often leads to slightly worse results compared to its backbones [5], there are cases where interpretable models outperform their non-interpretable counterparts [6, 23]. This phenomenon can also be observed in this experiment, where the BagNet–17 model shows the highest values across all performance measures. This could either be caused by the simplicity of the binary classification problem, or the different nature of tissue scans compared to natural images from ImagNet.

3.3 Qualitative evaluation and interpretability

As our approach is intended to replace human specialists in the safety-critical healthcare industry, one of the main reasons for choosing the BagNet archi-

Model	Acc.	Prec.	Recall	Spec.
Random weights				
VGG19	0.8	0.625	0.5	0.9
ResNet50	0.925	0.8182	0.9	**0.9333**
BagNet–17	**0.95**	**0.8333**	**1.0**	**0.9333**
ImageNet weights				
VGG19	0.925	0.7692	**1.0**	0.9
ResNet50	0.95	0.8333	**1.0**	0.9333
BagNet–17	**0.975**	**0.9091**	**1.0**	**0.9667**

Table 5: Results of training non-interpretable CNNs for the binary classification task. The values shown in this table are the average validation accuracy, precision, recall, and specificity scores obtained after training the models on the LUAD – NEC dataset with and without using data augmentation.

tecture was its inherent explainability. To demonstrate expected behavior, we hereby evaluate the decision-making process of our models, which in the case of the adapted BagNet architecture consists in the analysis of the models' pixel activation heatmaps. As all of our datasets contain expert annotated segmentation masks for both validation and testing, the exact location of tissue types on WSIs can be leveraged to examine the validity of heatmap pixel activations, thus making it possible to quantitatively measure model interpretability and accordance with expected behavior.

In this experiment, we evaluated trained models on specific WSI patches from the test subsets of different binary datasets and measured in what amount the decisions were based on expected premises. This was done by identifying the top 5% most important heatmap patches, i.e. the ones which contribute most in inferring class evidence, and calculating what percentage of these were located inside the corresponding segmentation mask. Representative examples for this evaluation are shown in Figures 4 and 5. Both of these figures illustrate model performance using four images divided clockwise as follows: the model input WSI patch is visible on the upper left, the segmentation mask for the specific class is shown in the upper right image, the lower right image shows the pixel activation heatmaps for the trained model with a pale outline of the segmentation mask, and the image on the lower left illustrates the top 5% most influential pixels colored green if they lie inside the segmentation mask and yellow otherwise. The most important heatmap patches are illustrated in red.

Specific examples from datasets of different sizes are shown in Figure 4. The difference in WSI qualities between the LUAD and BCSS datasets is visible when comparing segmentation masks of the BCSS – STR example with the other examples from the LUAD datasets. The magnification of the BCSS samples appears to be significantly greater, thus resulting in less detailed samples and segmentation masks as well. This observation should be taken into account when evaluating BagNets' performance on this dataset, which although showing 67.26% of the most important patches in their expected location, appear to be heavily concentrated in the edge of the sample.

When evaluating the other three examples from this figure, the BagNet models appear to be showing promising behavior. The examples from the LUAD – LYM and LUAD – TE datasets demonstrate a decision-making process that aligns with the expected segmentation masks with 80.38% and respectively 89% of the most important patches being inside of their expected locations. The LUAD – NEC example shows slightly weaker performance in this context, however, this might be caused by this class being inherently more difficult to identify than others. Therefore, because of the possible difference in classification difficulty for different datasets, conclusions can not be drawn about the influence of dataset size on the decision-making process of the models.

Model generalizability can be improved by augmenting training data [15]. Therefore, we examine the effects of data augmentation on the decision-making process of the BagNet models by analyzing Figure 5, where examples for two different classes are shown with and without data augmentation. By comparing the two heatmaps of any one of the two classes side-by-side, the confidence-increasing effect of using data augmentation becomes apparent. The patches on the examples without data augmentation are significantly more blurred than on the ones with augmentation, especially in the case of the LUAD – NEC example. This effect might be a consequence of the low diversity dataset expansion resulting from augmentation. Thus, the augmented dataset contains multiple slightly different versions of all images, which could inherently increase classification confidence.

In terms of the more quantitative metric of the percentage of most important patches in their expected location the effects of data augmentation are not as clear. In the case of the LUAD – NEC example, which showed the worst performance in Figure 4, data augmentation leads to significant improvements in terms of decision making with an increase from 39% to 62.64% of most important patches in their expected location. However, applying data augmentation on the smaller LUAD – LYM dataset leads to a slightly worse value for this metric, with an effect of moderately spreading the most important

patches over the image. Apart from this minor decay in expected behavior, we overall advise the usage of data augmentation for the task of TME tissue binary classification because of the effects of increased confidence and accuracy improvements shown in Section 3.2.

4 Conclusion and discussion

The automation of tasks performed by human pathologists poses various challenges. For instance, a machine learning model trained on data from one hospital may perform poorly on data from another hospital due to differences in scanning equipment and tissue processing protocols. Medical datasets also require expert annotation, which is time-consuming and resource-intensive, leading to a scarcity of high-quality annotated data. Moreover, the black-box nature of deep learning models is also a significant challenge in medical image processing, where the interpretability and explainability of algorithms are becoming essential.

The focus of this paper lies in solving the cancer detection problem of identifying differences in tissue types of tumor microenvironments. Our approach to TME tissue type detection involves breaking it down into multiple binary classification problems. This method can be used in conjunction with or as a replacement for human radiologists to label whole-slide images, which can later be used as inputs for more complex segmentation techniques.

To accomplish binary tissue classification, we introduced a modified version of the explainable BagNet architecture. The proposed model is capable of outperforming human experts in different binary classification tasks, providing a viable alternative to human-based tissue detection. Using the BagNet network architecture also addresses concerns surrounding the back-box nature of medical image processing models by being able to justify its decisions. In addition, we also demonstrated that the decision-making process of these models is also aligned with that of human radiologists. Furthermore, we addressed the challenge of limited medical data by augmenting our binary datasets, and we analyzed the effects of augmentation techniques on model performance and decision-making validity.

Being able to accurately and reliably identify TME tissue types without the need for the involvement of human experts presents a variety of different advantages in the field of medical imaging. Firstly, it leads to the burden of disease diagnosis being taken off the shoulders of pathologists, them being able to focus on patient treatment and care. Secondly, this could also lead to new

developments in more complex classification and segmentation tasks regarding tumor microenvironments. For instance, our trained models could also be used for creating large datasets annotated with patch-level classification labels of tissue types, thus providing future models with sufficiently large annotated datasets without the necessity for human involvement. Moreover, these models could provide a way to fully automate more complex approaches that rely on human labeling, such as the method introduced in [11].

The first future research direction possibly worth exploring regarding the proposed models is the evaluation of the benefits these create in terms of new developments in this field. This could be done by using the models to create annotated datasets, which would then be used for training and benchmarking various state-of-the-art approaches in TME image processing. The usability of these models as initial steps for more complex methods should also be evaluated by carrying out experiments comparing performances with and without using them. Another related question concerns the robustness of the models for WSI differences caused by artifacts, discoloration, and tissue deterioration. This characteristic could be reviewed by evaluating the models on new datasets obtained at different laboratories with slight variations in tissue sample acquisition pipelines.

Further research regarding this work could be carried out in the direction of model decision analysis. Although heatmaps can be reviewed qualitatively, there is a requirement for a more objective quantitative evaluation. The process of location of the most relevant patches carried out in this paper represented one quantitative evaluation method, however, more complex and representative quantities could also be analyzed, such as the relevance mass accuracy and relevance rank accuracy metrics proposed in [3]. Moreover, model performance could be evaluated on larger validation and testing datasets as well to further solidify results regarding classification accuracy.

Acknowledgements

This work was supported by Babeş–Bolyai University's Special Scholarship for Scientific Activity (No. 36586/25.11.2022) and the Collegium Talentum Programme of Hungary. A part of the research leading to these results was supported by the Márton Áron College of ELTE Eötvös Loránd University. We thank Dr. Bognár Gergő and Dr. Kovács Péter from the Faculty of Numerical Analysis, ELTE Eötvös Loránd University for providing the computer hardware used in the experiments.

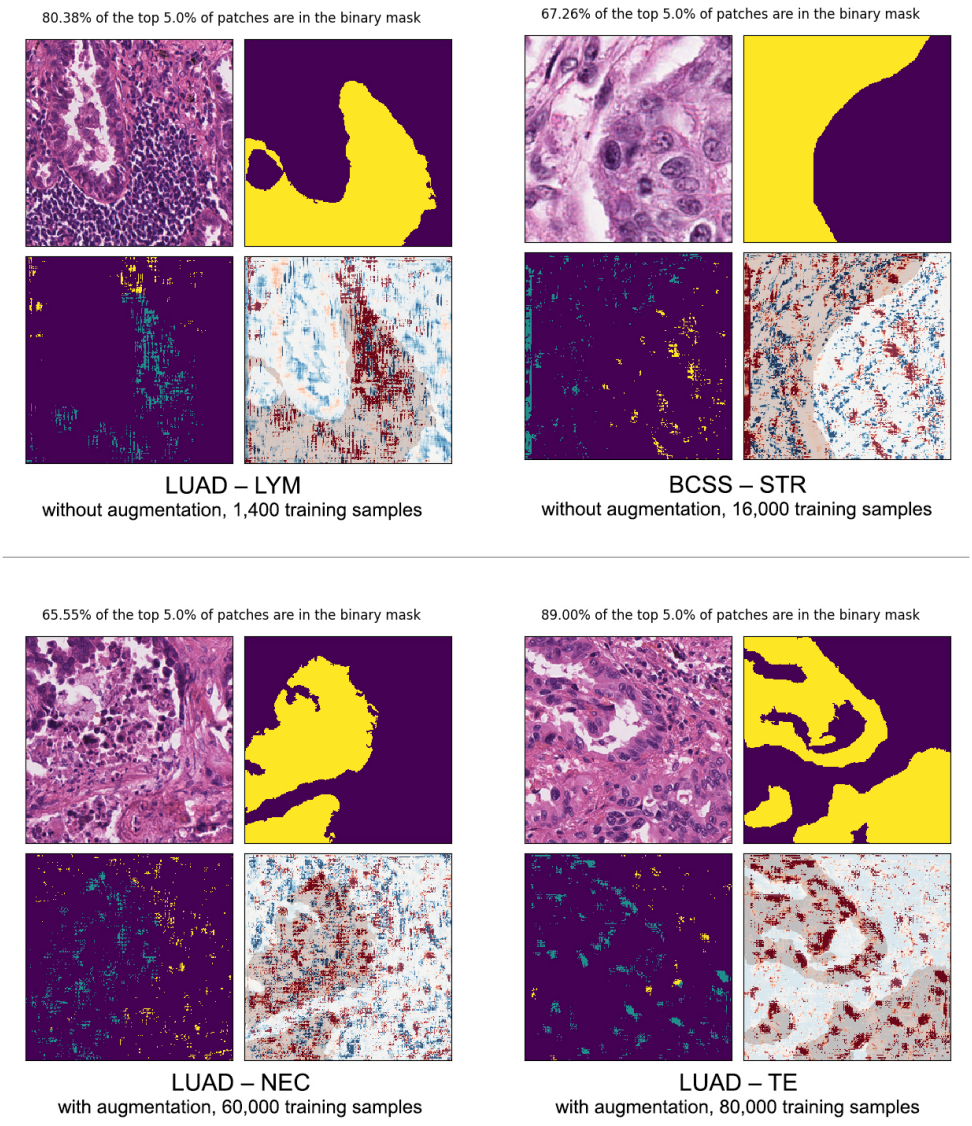


Figure 4: Heatmap analysis of BagNet models trained on four different datasets with varying sizes.

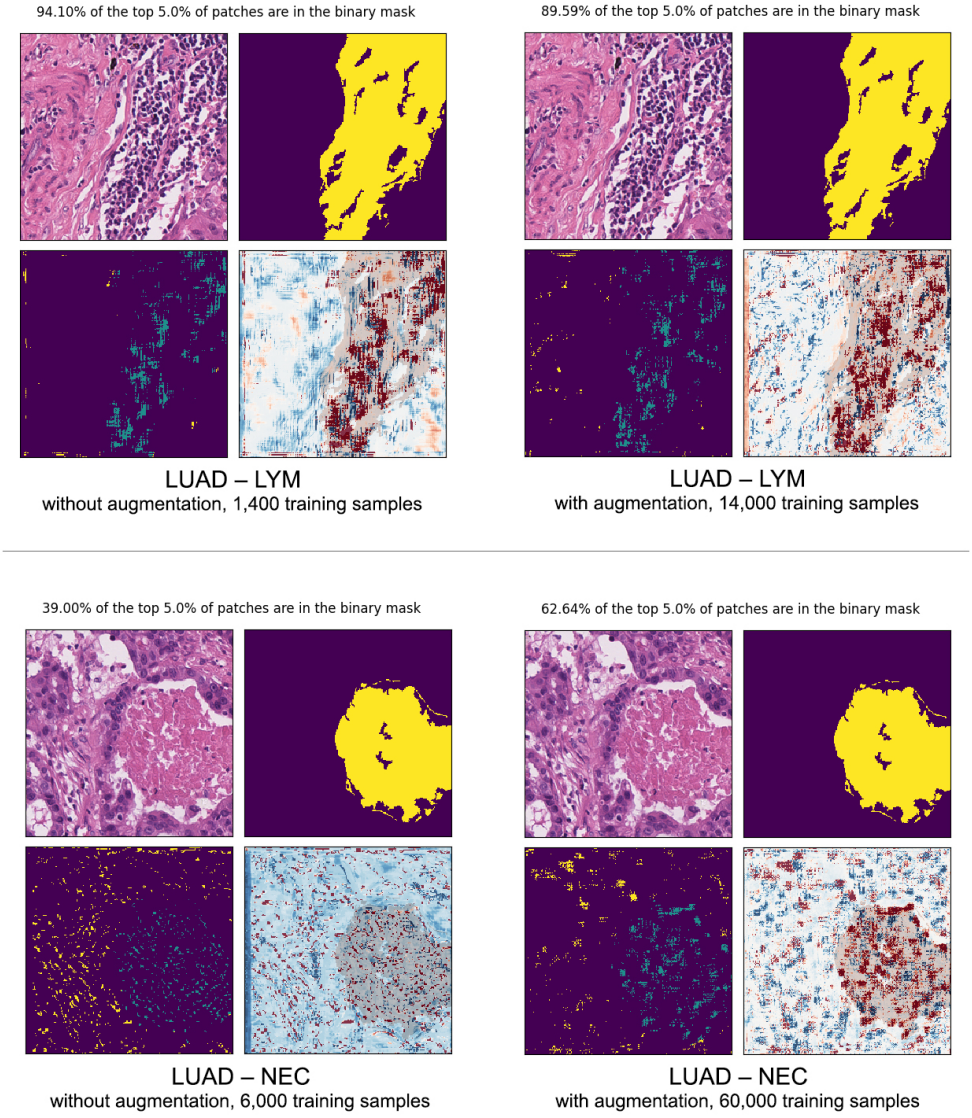


Figure 5: Heatmap analysis of BagNet models trained for binary classification of two different classes with and without data augmentation.

References

[1] K. AbdulJabbar, S. E. A. Raza, R. Rosenthal, M. Jamal-Hanjani, S. Veeriah, A. Akarca, T. Lund, D. A. Moore, R. Salgado, M. Al Bakir, L. Zapata, Geospatial immune variability illuminates differential evolution of lung adenocarcinoma, *Nature Medicine* 26(7), 2020, pp. 1054–1062. ⇒61

[2] M. Amgad, H. Elfandy, H. Hussein, L. A. Atteya, M. A. Elsebaie, L. S. Abo Elnasr, R. A. Sakr, H. S. Salem, A. F. Ismail, A. M. Saad, J. Ahmed, Structured crowdsourcing enables convolutional segmentation of histology images, *Bioinformatics* 35(18), 2019, pp. 3461–3467. ⇒61, 63, 64

[3] L. Arras, A. Osman, W. Samek, CLEVR-XAI: a benchmark dataset for the ground truth evaluation of neural network explanations, *Information Fusion* 81, 2022, pp. 14–40. ⇒76

[4] M. S. Ayhan, L. B. Kümmerle, L. Kühlewein, W. Inhoffen, G. Aliyeva, F. Ziemssen, P. Berens, Clinical validation of saliency maps for understanding deep neural networks in ophthalmology, *Medical Image Analysis* 77, 2022, p. 102364. ⇒68

[5] W. Brendel, M. Bethge, Approximating CNNs with bag-of-local-features models works surprisingly well on ImageNet, *arXiv preprint* arXiv:1904.00760, 2019. ⇒62, 63, 66, 67, 68, 69, 72

[6] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, J. K. Su, This looks like that: deep learning for interpretable image recognition, *Advances in neural information processing systems* 32, 2019. ⇒72

[7] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248-255). ⇒63

[8] I. B. Dimenstein, Grossing biopsies: an introduction to general principles and techniques, *Annals of Diagnostic Pathology* 13(2), 2009, pp. 106–113. ⇒61

[9] K. Doi, Current status and future potential of computer-aided diagnosis in medical imaging, *The British Journal of Radiology* 78, 2005, pp. 3–19. ⇒61

[10] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, T. Chen, Recent advances in convolutional neural networks, *Pattern Recognition* 77, 2018, pp. 354–377. ⇒67

[11] C. Han, J. Lin, J. Mai, Y. Wang, Q. Zhang, B. Zhao, X. Chen, X. Pan, Z. Shi, Z. Xu, S. Yao, Multi-layer pseudo-supervision for histopathology tissue semantic segmentation using patch-level classification labels, *Medical Image Analysis* 80, 2022, p. 102487. ⇒61, 62, 63, 71, 76

[12] D. Hanahan, R. A. Weinberg, Hallmarks of cancer: the next generation, *Cell* 144(5), 2011, pp. 646–674. ⇒61

[13] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. ⇒67

[14] L. Hou, D. Samaras, T. M. Kurc, Y. Gao, J. E. Davis, J. H. Saltz, Patch-based convolutional neural network for whole slide tissue image classification, *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2424–2433. ⇒72

[15] N. Kanwal, F. Pérez-Bueno, A. Schmidt, K. Engan, R. Molina, The devil is in the details: whole slide image acquisition and processing for artifacts detection, color variation, and data augmentation: a review, *IEEE Access* 10, 2022, pp. 58821–58844. ⇒61, 74

[16] S. Morales, K. Engan, V. Naranjo, Artificial intelligence in computational pathology -– challenges and future directions, *Digital Signal Processing* 119, 2021, p. 103196. ⇒61

[17] C. Park, H. I. Suk, Deep joint learning of pathological region localization and Alzheimer's disease diagnosis, *arXiv preprint* arXiv:2108.04555, 2021. ⇒63, 68

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, ImageNet large scale visual recognition challenge, *International Journal of Computer Vision* 115, 2015, pp. 211–252. ⇒63

[19] J. Su, D. V. Vargas, K. Sakurai, One pixel attack for fooling deep neural networks, *IEEE Transactions on Evolutionary Computation* 23(5), 2019, pp. 828–841. ⇒61

[20] A. Taleb, W. Loetzsch, N. Danz, J. Severin, T. Gaertner, B. Bergner, C. Lippert, 3D self-supervised methods for medical imaging, *Advances in Neural Information Processing Systems* 33, 2020, pp. 18158–18172. ⇒61, 63

[21] E. Tjoa, C. Guan, A survey on explainable artificial intelligence (XAI): Toward medical XAI, *IEEE Transactions on Neural Networks and Learning Systems* 32(11), 2020, pp. 4793–4813. ⇒61

[22] B. H. Van der Velden, H. J. Kuijf, K. G. Gilhuijs, M. A. Viergever, Explainable artificial intelligence (XAI) in deep learning-based medical image analysis, *Medical Image Analysis*, 2022, p. 102470. ⇒61

[23] J. Wang, H. Liu, X. Wang, L. Jing, Interpretable image recognition by constructing transparent embedding space, *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 895-904. ⇒72

[24] M. Wang, D. Zhang, D. Shen, M. Liu, Multi-task exclusive relationship learning for Alzheimer's disease progression prediction with longitudinal data, *Medical Image Analysis* 53, 2019, pp. 111–122 ⇒61

[25] C. Wang, X. Zhu, J. C. Hong, D. Zheng, Artificial intelligence in radiotherapy treatment planning: present and future, *Technology in Cancer Research & Treatment* 18, 2019, p. 1533033819873922. ⇒61

Received: May 9, 2023 • Revised: June 19, 2023

Acta Univ. Sapientiae Informatica **15,** 1 (2023) 81–90

DOI: 10.2478/ausi-2023-0007

E-super arithmetic graceful labelling of $H_i(m,m)$, $H_i^{(1)}(m,m)$ and chain of even cycles

S. ANUBALA
Department of Mathematics,
Sri Kaliswari College
(Autonomous),Sivakasi,
Virudhunagar, TamilNadu, India.
email: anubala.ias@gmail.com

V. RAMACHANDRAN
PG and Researh Department of
Mathematics,
Mannar Thirumalai Naicker
College(Autonomous), Pasumalai,
Madurai, TamilNadu,India.
email: me.ram111@gmail.com

Abstract. E-super arithmetic graceful labelling of a graph G is a bijection f from the union of the vertex set and edge set to the set of positive integers $(1,2,3,\ldots|V(G)\cup E(G)|)$ such that the edges have the labels from the set $\{1,2,3,\ldots,|E(G)|\}$ and the induced mapping f^* given by $f^*(uv) = f(u) + f(v) - f(uv)$ for $uv \in E(G)$ has the range $\{|V(G)\cup E(G)|+1, |V(G)\cup E(G)|+2,\ldots,|V(G)|+2|E(G)|\}$

In this paper we prove that $H_i(m,m)$ and $H_i^{(1)}(m,m)$ and chain of even cycles $C_{4,n}$, $C_{6,n}$ are E-super arithmetic graceful.

1 Introduction

Rosa [9] in 1967, called a function f a β-valuation of a graph G with q edges if f is an injection from the vertices of G to the set $\{0,1,\ldots,q\}$ such that when each edge xy is assigned the label $|f(x)-f(y)|$, the resulting edge labels are distinct. Golomb [3] subsequently called such labelling graceful.

Key words and phrases: E-super,$H_i(m,m)$, $H_i^{(1)}(m,m)$, $C_{4,n}$, $C_{6,n}$.

In 1970 Kotzig and Rosa [5] defined a *magic valuation of a graph* $G(V,E)$ as a bijection f from $V \cup E$ to $\{1,2,\ldots,|V \cup E|\}$ such that for all edges xy, $f(x)+f(y)+f(xy)$ is constant (called the magic constant).

Acharya and Hegde [1] have defined (k,d)-arithmetic graphs.
Let G be a graph with q edges and let k and d be positive integers. A labelling f of G is said to be (k,d)-arithmetic if the vertex labels are distinct nonnegative integers and the edge labels induced by $f(x)+f(y)$ for each edge xy are $k, k+d, k+2d, \ldots, k+(q-1)d$. The case where $k=1$ and $d=1$ was called additively graceful by Hegde [4].

J. A. Gallian [2] surveyed numerous graph labelling methods.

V. Ramachandran and C. Sekar [8] introduced (1,N)-arithmetic labelling.

A labelling of $G(V,E)$ is said to be E-super if $f(E(G)) = \{1,2,3,\ldots,|E(G)|\}$.

MacDougall, Slamin, Miller and Wallis [6] introduced the notion of a vertex-magic total labelling in 1999. For a graph $G(V,E)$ an injective mapping f from $V \cup E$ to the set $\{1,2,\ldots,|V|+|E|\}$ is a vertex-magic total labeling if there is a constant k, called the magic constant such that for every vertex v, $f(u)+\Sigma f(vu) = k$ where the sum is taken over all vertices u adjacent to v.

Marimuthu and Balakrishnan [7] defined a graph $G(V,E)$ to be edge magic graceful if there exists a bijection f from $V(G) \cup E(G)$ to $\{1,2,\ldots,p+q\}$ such that $|f(u)+f(v)-f(uv)|$ is a constant for all edges uv of G.

We define a graph $G(p,q)$ to be ***E-super arithmetic graceful*** if there exists a bijection f from $V(G) \cup E(G)$ to $\{1,2,\ldots,p+q\}$ such that $f(E(G)) = \{1,2,\ldots,q\}$, $f(V(G)) = \{q+1, q+2, \ldots, q+p\}$ and the induced mapping f^* given by $f^*(uv) = f(u)+f(v)-f(uv)$ for $uv \in E(G)$ has the range $\{p+q+1, p+q+2, \ldots, p+2q\}$.

In this paper we prove that $H_i(m,m)$ and $H_i^{(1)}(m,m)$ and $C_{4,n}$,$C_{6,n}$ are E-super arithmetic graceful.

2 Preliminaries

Definition 1 *A connected graph is* ***highly irregular*** *if each of its vertices is adjacent only to vertices with distinct degrees. Let* H *denote the bipartite graph of order* $n = 2m$, $m \geq 2$ *having partite sets,* $V_1 = \{u_1, u_2, \ldots, u_m\}$ *and* $V_2 = \{v_1, v_2, \ldots, v_m\}$ *and edge set* $E(H) = \{u_i v_j : 1 \leq i \leq m, 1 \leq j \leq m+1-i\}$ *with* $\deg_H(u_i) = \deg_H(v_i) = m+1-i$ *for* $i = 1,2,\ldots,m$.
H *is a irregular graph of order* $n = 2m$, $m \geq 2$. *Let us denote this graph as* $H_i(m,m)$.

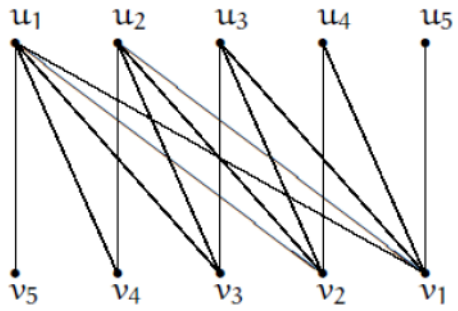


Figure 1: $H_i(5,5)$ – highly irregular graph of order 10

Definition 2 *By subdividing the edge* u_2v_{m-1} *of* $H_i(m,m)$ *for* $m \geq 4$*, we obtain a highly irregular graph of order* $2m+1 \geq 9$*. Denote this graph by* $H_i^{(1)}(m,m)$*.*

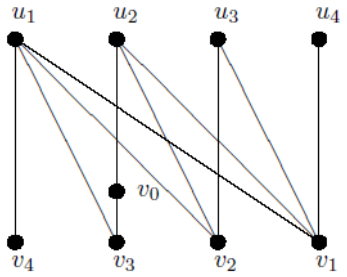


Figure 2: $H_i^{(1)}(4,4)$ – highly irregular graph of order 9

Definition 3 *Let* C_{2k} *be an even cycle. Consider* n *copies of* C_{2k}*. A chain of even cycles* C_{2k} *denoted by* $C_{2k,n}$ *is obtained by identifying the vertex* u_{k+1} *of each copy of* C_{2k} *with the vertex* u_1 *of the successive copy of* C_{2k}*.*
$C_{2k,n}$ *has* $(2k-1)n+1$ *vertices and* $2kn$ *edges.*
$C_{2k,n}$ *has* $(k-1)n$ *upper nodes* $u_1, u_2, \ldots, u_{(k-1)n}$ *;*
$(k-1)n$ *lower nodes* $w_1, w_2, \ldots, w_{(k-1)n}$ *and* $(n+1)$ *middle nodes* $v_1, v_2, \ldots, v_{n+1}$*.*

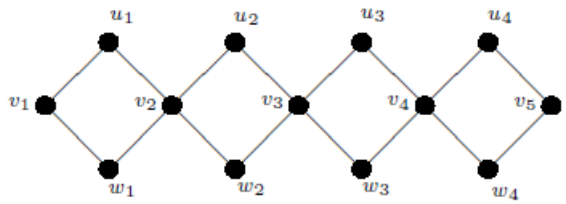


Figure 3: $C_{4,4}$

3 Main results

Theorem 4 *$H_i(m, m)$ is E-super arithmetic graceful for $m \geq 2$.*

Proof. Let $G = H_i(m, m)$. G has $2m$ vertices and $\binom{m+1}{2} = \frac{m(m+1)}{2}$ edges.
Define $f : V(G) \cup E(G) \longrightarrow \left\{1, 2, \ldots, 2m + \frac{m(m+1)}{2}\right\}$ as follows:

$$f(u_i) = \binom{m+1}{2} + i, \qquad i = 1, 2, \ldots, m.$$

$$f(v_i) = \binom{m+1}{2} + 2m + 1 - i, \quad i = 1, 2, \ldots, m$$

$$f(u_i v_j) = \binom{m+1}{2} + i - \binom{i+j}{2},\ i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, (m+1) - i$$

Clearly f is a bijection from $V \cup E$ to $\left\{1, 2, \ldots, \binom{m+1}{2} + 2m\right\}$ where $f(E) = \left\{1, 2, \ldots, \binom{m+1}{2}\right\}$.
Also

$$f^*(E(H_i(m, m))) = \left\{\binom{m+1}{2} + 2m + 1, \binom{m+1}{2} + 2m + 3, \ldots, 2\left[\binom{m+1}{2} + m\right]\right\}$$

Therefore, $H_i(m, m)$ for $m \geq 2$ is E-super arithmetic graceful. □

Example 5 *E-super arithmetic graceful labelling of* $H_i(6,6)$.

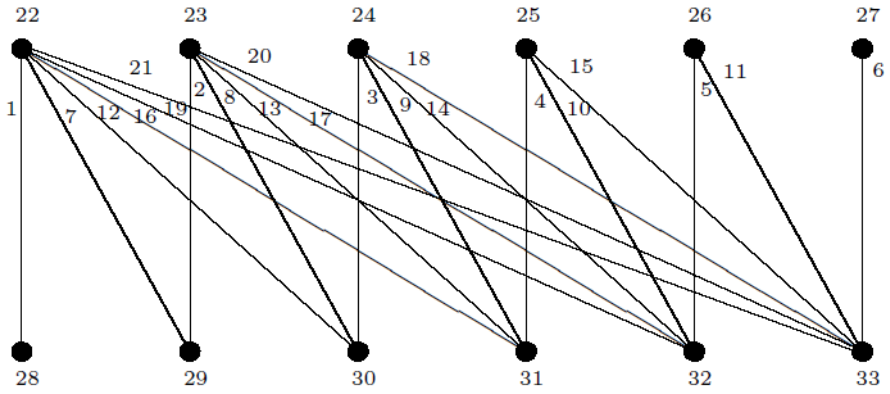


Figure 4: $H_i(6,6)$

Theorem 6 $H_i^{(1)}(m,m)$ *for* $m \geq 4$ *is E-super arithmetic graceful.*

Proof. $H_i^{(1)}(m,m)$ for $m \geq 4$ has $2m+1$ vertices and $\binom{m+1}{2}+1$ edges.
Define

$$f(u_i) = \binom{m+1}{2} + 1 + i, \quad i = 1,2,\ldots,m.$$

$$f(v_0) = \binom{m+1}{2} + m + 2,$$

$$f(v_i) = \binom{m+1}{2} + 2m + 3 - i, \quad i = 1,2,\ldots,m$$

$f(u_1v_1) = 1$
$f(u_1v_m) = 2$
$f(u_2v_0) = 3$
$f(v_0v_{m-1}) = m+2$
$f(u_iv_{m+1-i}) = i+1, \quad i = 3,4,\ldots m$
For $i = 1$; $\quad j = 2,3,\ldots,m-1$ and
for $i = 2,3,\ldots,m, \quad j = 1,2,\ldots,(m+1)-i$

$$f(u_iv_j) = \binom{m+1}{2} + 2 + i - \binom{i+j}{2}$$

Clearly f is a bijection from $V \cup E$ to $\left\{1,2,\ldots,\binom{m+1}{2}+2m+2\right\}$ where

$$f(E) = \left\{1, 2, \ldots, \binom{m+1}{2} + 1\right\}.$$

Also

$$f^*(E(H_i^{(1)}(m,m))) = \left\{\binom{m+1}{2} + 2m + 3, \binom{m+1}{2} + 2m + 5, \ldots, 2\left[\binom{m+1}{2} + m\right] + 3\right\}$$

Therefore, $H_i^{(1)}(m,m)$ for $m \geq 4$ is E-super arithmetic graceful. □

Example 7 *E-super arithmetic graceful labelling of* $H_i^{(1)}(5,5)$.

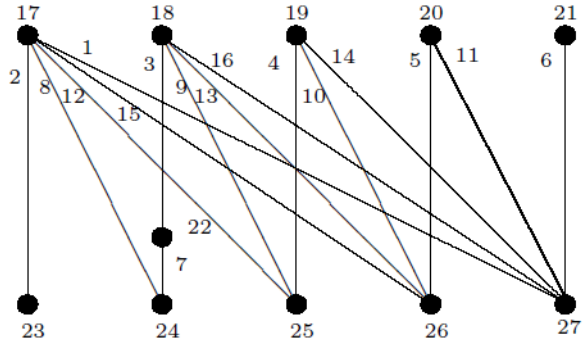


Figure 5: $H_i^{(1)}(5,5)$

Theorem 8 $C_{4,n}$ *are E-super arithmetic graceful.*

Proof. $C_{4,n}$ has $3n+1$ vertices and $4n$ edges.
Let $u_1, u_2, \ldots, u_n$ be the upper nodes,$w_1, w_2, \ldots, w_n$ be the lower nodes and $v_1, v_2, \ldots, v_{n+1}$ be the middle nodes.
Define $f(u_i) = 4n + i, \quad i = 1, 2, \ldots, n.$
$f(v_i) = 5n + i, \quad i = 1, 2, \ldots, n+1.$
$f(w_i) = 6n + 1 + i, \quad i = 1, 2, \ldots, n.$
$f(u_1v_1) = n + 1$
$f(u_iv_i) = 2n + i, \quad i = 2, \ldots, n$
$f(u_iv_{i+1}) = i, \quad i = 1, 2, \ldots, n$
$f(v_iw_i) = 3n + i, \quad i = 1, 2, \ldots, n$

$f(v_{i+1}w_i) = n + 1 + i, \quad i = 1, 2, \ldots, n$
$f^*(u_1v_1) = 8n + 1$

$$\{f^*(u_iv_i) \mid i = 1, 2, 3, \ldots, n\} = \{7n + 2, 7n + 3, \ldots, 8n\}$$
$$\{f^*(u_iv_{i+1}) \mid i = 1, 2, 3, \ldots, n\} = \{9n + 2, 9n + 3, \ldots, 10n + 1\}$$
$$\{f^*(v_iw_i) \mid i = 1, 2, 3, \ldots, n\} = \{8n + 2, 8n + 3, \ldots, 9n + 1\}$$
$$\{f^*(v_{i+1}w_i) \mid i = 1, 2, 3, \ldots, n\} = \{10n + 2, 10n + 3, \ldots, 11n + 1\}$$

Thus $f^*(E(C_{4,n})) = \{7n + 2, 7n + 3, \ldots, 11n + 1\}$.
Therefore, $C_{4,n}$ is E-super arithmetic graceful. □

Example 9 *E-super arithmetic graceful labelling of* $C_{4,5}$.

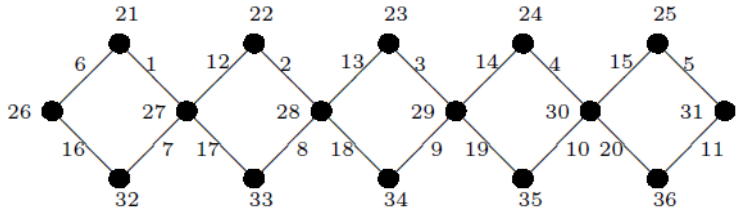


Figure 6: $C_{4,5}$

Theorem 10 $C_{6,n}$ *is E-super arithmetic graceful for all* n.

Proof. Let $G = C_{6,n}$. Let $u_1^{(1)}, u_1^{(2)}, u_2^{(1)}, u_2^{(2)}, \ldots, u_n^{(1)}, u_n^{(2)}$ be the upper level vertices of G.
Let $w_1, w_2, \ldots, w_{n+1}$ be the middle level vertices of G.
Let $v_1^{(1)}, v_1^{(2)}, v_2^{(1)}, v_2^{(2)}, \ldots, v_n^{(1)}, v_n^{(2)}$ be the upper level vertices of G.

Illustration: $C_{6,4}$

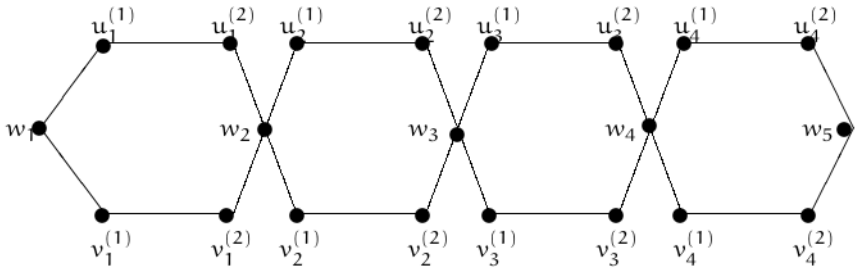


Figure 7: $C_{6,4}$

$C_{6,n}$ has 5n+1 vertices and 6n edges.
Define $f : V(G) \cup E(G) \longrightarrow \{1, 2, 3, \ldots, 11n + 1\}$ as follows:
$f(u_i^{(1)}) = 6n + i, \quad i = 1, 2, \ldots, n$
$f(u_i^{(2)}) = 7n + i, \quad i = 1, 2, \ldots, n$
$f(w_i) = 8n + i, \quad i = 1, 2, \ldots, n + 1$
$f(v_i^{(1)}) = 9n + 1 + i, \quad i = 1, 2, \ldots, n$
$f(v_i^{(2)}) = 10n + 1 + i, \quad i = 1, 2, \ldots, n$
$f(u_i^{(1)}u_i^{(2)}) = i, \quad i = 1, 2, \ldots, n$
$f(u_i^{(2)}w_{i+1}) = n + i, \quad i = 1, 2, \ldots, n$
$f(w_i u_i^{(1)}) = 3n - 1i, \quad i = 1, 2, \ldots, n$
$f(v_i^{(2)}w_{i+1}) = 2n + i, \quad i = 1, 2, \ldots, n - 1$
$f(v_i^{(1)}v_i^{(2)}) = 4n - 1 + i, \quad i = 1, 2, \ldots, n - 1$
$f(v_n^{(1)}v_n^{(2)}) = 6n$
$f(w_i v_i^{(1)}) = 5n + i, \quad i = 1, 2, \ldots, n - 1$
$f(w_n v_n^{(1)}) = 5n$
$f(v_n^{(2)}w_{n+1}) = 5n - 1$
Clearly f is a bijection and $f(E(G)) = \{1, 2, \ldots, 6n\}$
$\left\{f^*(u_i^{(1)}u_i^{(2)}) \mid i = 1, 2, \ldots, n\right\} = \{13n + 1, 13n + 2, \ldots, 14n\}$
$\left\{f^*(u_i^{(2)}w_{i+1}) \mid i = 1, 2, \ldots, n\right\} = \{14n + 2, 14n + 3, \ldots, 15n + 1\}$
$\left\{f^*(w_i u_i^{(1)}) \mid i = 1, 2, \ldots, n\right\} = \{11n + 2, 11n + 3, \ldots, 12n + 1\}$
$\left\{f^*(w_i v_i^{(1)}) \mid i = 1, 2, \ldots, n - 1\right\} = \{12n + 2, 12n + 3, \ldots, 13n\}$

$\left\{f^*(v_i^{(2)}w_{i+1}) \mid i = 1, 2, \ldots, n-1\right\} = \{16n+3, 16n+4, \ldots, 17n+1\}$
$\left\{f^*(v_i^{(1)}v_i^{(2)}) \mid i = 1, 2, \ldots, n-1\right\} = \{15n+4, 15n+5, \ldots, 16n+2\}$
$\left\{f^*(w_n v_n^{(1)})\right\} = 14n+1$
$\left\{f^*(v_n^{(1)}v_n^{(2)})\right\} = 15n+2$
$\left\{f^*(v_n^{(2)}w_{n+1})\right\} = 15n+3$
Combining all the above we have $f^*(E(G)) = \{11n+2, 11n+3, \ldots, 17n+1\}$
Therefore, G is E-super arithmetic graceful. □

Example 11 *E-super arithmetic graceful labelling of* $C_{6,6}$.

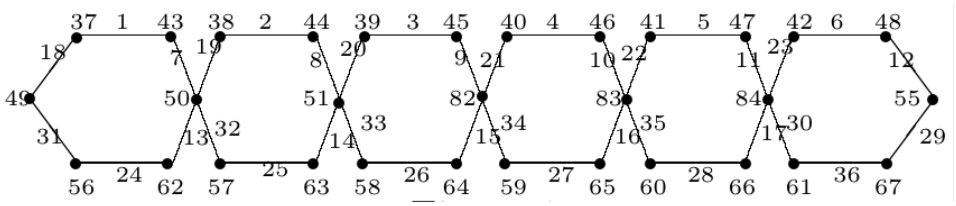


Figure 8: $C_{6,6}$

Conjecture: *Chains of all even cycles* $C_{2m,k}$ *are E-super arithmetic graceful.*

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the article.

References

[1] B. D. Acharya, S. M. Hedge, Arithmetic graphs, *J. Graph Theory*, **14** (1990) 275–299. ⇒82
[2] J. A. Gallian, A dynamic survey of graph labelling, *The Electronic Journal of combinatorics*, DS6 (2016). ⇒82
[3] S. W. Golomb, How to number a graph, in *Graph Theory and Computing*, R. C. Reed (ed) Academic Press, New York (1972) 23–37. ⇒81
[4] S. M. Hedge, Additively graceful graphs, *Mat. Acad. Sci. Lett.*, **12** (1989) 387–390. ⇒82

[5] A. Kotzig, A. Rosa Magic valuation of finite graphs, *Canad. Math. Bull*, **13** (1970) 451–456. ⇒82

[6] J. A. MacDougall, M. Miller, Slamin, W. D. Walls, Vertex-magic total labelling of graphs, *Util. Math.* **61** (2002) 3–21. ⇒82

[7] G. Marimuthu, M. Balakrishnan, Super edge magic graceful graphs, *Information Sciences*, **287,** 10 (2014) 140–151. http://dx.doi.org/10.10162Fj.ins.2014.07.027 ⇒82

[8] V. Ramachandran, C. Sekar, $(1, \mathbb{N})$-arithmetic graphs, *International Journal of Computers and Applications*, Vol. 38, 1 (2016) 55–59. https://doi.org/10.1080/1206212X.2016.1218240 ⇒82

[9] A. Rosa, On certain valuations of the vertices of a graph, *Theory of graphs* (International Symposium, Rome, July 1966), Gordon and Breach, N.Y and Dunod Paris (1967) 349–355. ⇒81

Received: January 18, 2022 • Revised: June 19, 2023

 Acta Univ. Sapientiae Informatica **15,** 1 (2023) 91–110

DOI: 10.2478/ausi-2023-0008

Textual outlier detection with an unsupervised method using text similarity and density peak

Mahnaz TALEB SERESHKI
Computer Engineering Department, Imam Khomeini International University, Qazvin, Iran
email: M.talebsereshki@edu.ikiu.ac.ir

Morteza MOHAMMADI ZANJIREH[a]
Computer Engineering Department, Imam Khomeini International University, Qazvin, Iran
email: Zanjireh@eng.ikiu.ac.ir

Mahdi BAHAGHIGHAT
Computer Engineering Department, Imam Khomeini International University, Qazvin, Iran
email: Bahaghighat@eng.ikiu.ac.ir

[a]Corresponding author

Abstract. Text mining is an intriguing area of research, considering there is an abundance of text across the Internet and in social medias. Nevertheless outliers pose a challenge for textual data processing. The ability to identify this sort of irrelevant input is consequently crucial in developing high-performance models. In this paper, a novel unsupervised method for identifying outliers in text data is proposed. In order to spot outliers, we concentrate on the degree of similarity between any two documents and the density of related documents that might support integrated clustering throughout processing. To compare the effectiveness of our proposed approach with alternative classification techniques, we performed a number of experiments on a real dataset. Experimental findings demonstrate that the suggested model can obtain accuracy greater than 98% and performs better than the other existing algorithms.

Key words and phrases: outlier detection, text data processing, unsupervised learning

1 Introduction

Countless specialists in data analysis are required for the precise processing and interpretation of data [21, 8, 33, 13, 37, 3]. In contrast, the influence of the Internet of Things (IoT) and social media has led to an enormous rise in the accumulation of data [13], [34]. Generally, data generation is prone to noise and unwanted changes. Since there is a lot of dirty data due to misinformation, disinformation, or bugs in data gathering, storage or call procedures, data cleansing can aid data analysts in achieving their objectives to create high accuracy models. Finding outliers is a proactive data-cleaning method. It is defined as an algorithm which tries to probe abnormal data [23]. Abnormal data represents information that deviates from the dataset's predominant patterns [22, 42]. In certain circumstances, system designers target finding outliers to study why these abnormal data are available in the system [38]. The significance of identifying outliers is demonstrated in credit fraud prevention and intrusion detection in computer networks [41, 24]. Because these sorts of complicated data need a lot of processing, spotting outliers in data can assist data scientists in improving the performance of their models [13, 24, 11]. Text data outliers can take on a variety of forms, and it might be challenging to identify them in this application.

On the identification of outliers in texts, numerous researchers have concentrated. As consequence, multiple aspects of the text's qualities are taken into account. A few, for instance, converted language into numbers and applied methods that are suitable for numerical data. Additionally, a number of research efforts exploited restricted phrases, such as document titles, to seek out patterns in datasets as well as recognize outliers. Plenty of studies have been conducted on dynamic datasets, such as social media, and the detection of anomalous data in such circumstances. It additionally addresses how significant knowledge is before any analysis ([23, 11, 20, 2, 30, 12, 35, 16]).

In this paper, our goal is to identify documents that do not follow the primary patterns of datasets in order to develop a high-performance text processing model from the total clusters that can be flexible in various situations, appropriate for text features, and implementation-unrestricted.

The remainder of the paper is structured as follows. The different kinds of methods for identifying outliers are discussed in Section 2. The solution we propose is elaborated in Section 3. Section 4 highlights our findings under various circumstances. Ultimately, Section 5 serves as the conclusion.

2 Related works

In general, there are several distinct types of outlier detection techniques. The classification-based algorithms are the subject of the first. Data in this situation ought to be labeled and utilized to train classification systems [17]. In contrast, the input to a clustering algorithm is not labeled, and the algorithm learns by solving problems similar to those it will face in the future as part of a training schedule. This makes a clustering algorithm an unsupervised model [28, 18].

Outlier detection is also possible with the clustering technique. Outlier detection based on clustering that comes following aims detection of "abnormal data". These algorithms identify outliers—those data that do not follow the typical data's patterns—as the data. Nearest neighbor is taken into account by several algorithms. The nearest neighbor plays a key role in these algorithms. Here, the term "abnormal data" refers to data that do not resemble their neighbors in any way [17, 25].

Other methods compute the probability of data being in a particular area employing probabilities and statistical models; if data exist in a region where this likelihood is low, the data is considered an outlier [17, 6, 5].

Subsequently, a few techniques are used for datasets that are dispersed throughout several platforms, like big data. These algorithms are made to look for outliers by taking into account the patterns of all the data in all systems [41, 31].

The remaining part of this section represents two well-known algorithms that we tested. One of them is a member of the clustering group, whereas the other one views the other as nearby neighbors.

2.1 Density-based spatial clustering of applications with noise (DBSCAN)

DBSCAN is one of the density-based clustering algorithms [10]. In this algorithm, two parameters are inputs (μ, ε), and points or data are categorized as core points, reachable points, and outliers. They are defined as follows:

- Core points are the points with at least μ neighbors in the border of ε from itself
- Reachable points are those points which are not core points but are located on the border of ε from one or more than one core point
- Outliers are those points that are not core or reachable points

Figure 1 demonstrates the differences between these kinds of points.

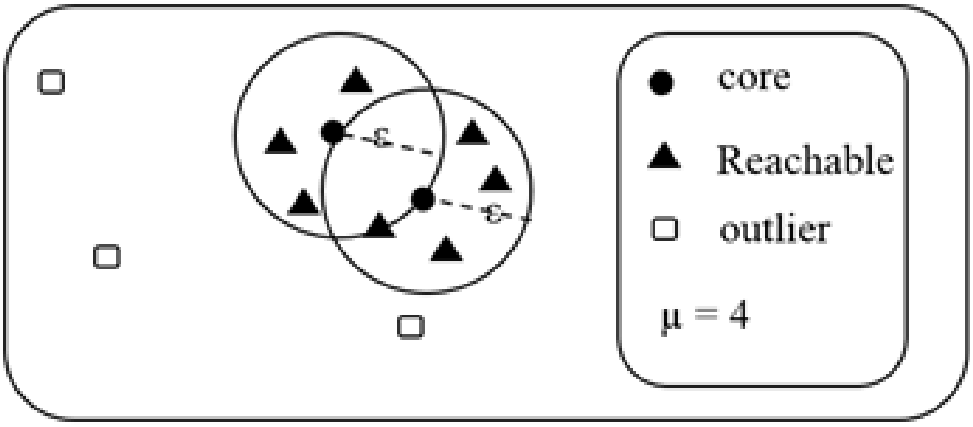


Figure 1: Core, reachable, and outlier points in DBSCAN

2.2 The improved Local Outlier Factor (LOF)

The goal of LOF which is based on the local density and nearest neighbors, is computing the local outlier factor which is the outliers' determination [40]. For computing the LOF of each point, calculating the reachability distance between the points and their k nearest neighbors, and also the Local Reachability Density (LRD) of the points are necessary. The reachability distance between two points (p, q) is defined as [12]:

$$reach\_dist(p, q) = max(k - distance(q), d(p, q)) \tag{1}$$

In Equation (1), $k - distance(q)$ is the distance between q and its k'th nearest neighbor, and $d(p, q)$ is the distance between p and q. The concept of reachability distance is shown in Figure 2.

And local reachability density of each point is defined as follows [12]:

$$LRDk(p) = K\frac{\sum_{q \in knn(p)} reach - dist(p.q)}{||k - neighborhood||} \tag{2}$$

In Equation 2, q is the neighbor of p, $reach - dist(p, q)$ is the reachability distance between p and q, and k-neighborhood is the number of p's neighbors. Finally, the Local Outlier Factor of each point is defined as [12]:

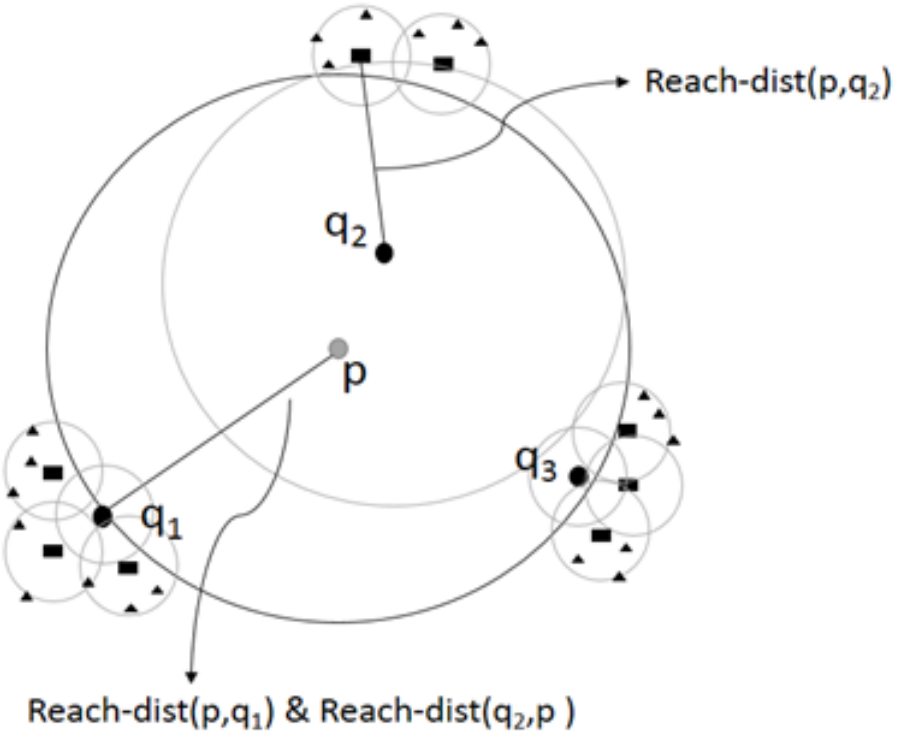


Figure 2: Reachability distance between two points [37]

$$\mathrm{LOFk}(p) = K\frac{\sum_{q\in knn(p)} \frac{\mathrm{LRD}(q)}{\mathrm{LRD}(p)}}{||k-neighborhood||} \tag{3}$$

In Equation 3, q is the neighbor of p, $\mathrm{LRD}(p)$ and $\mathrm{LRD}(q)$ are the local reachability density of p and local reachability density of q respectively. Furthermore, k-neighborhood is the number of p's neighbors. Whatever LOF is higher, the point is more abnormal.

3 Methodology

As we present in Section 2, the clustering algorithm is one of the groups of outlier detection, and this kind of algorithm can help to find abnormal data and the patterns of datasets. As mentioned before, the center of each cluster

is the point with a higher density than its neighbors and more different from the points with higher densities. In this case, the measurement of similarity is the distance between two points, but we prepare the algorithm for text data, therefore, cosine similarity is used. As a result, we propound local density for each document, denoting the number of text data or documents that are similar to this data. For this purpose, threshold and local density are considered as follows:

$$\rho_i = \sum_j X(t_i.t_j),\ X(a,b) = \begin{cases} 1, & \text{if } Similarity(a,b) > threshold \\ 0, & \text{if } Similarity(a,b) < threshold \end{cases} \tag{4}$$

In Equation 4, ρ_i is the local density of document i, and j is the number of the documents in a dataset, and t_j is an i_{th} document in the dataset. Figure 3 demonstrates computing ρ_i.

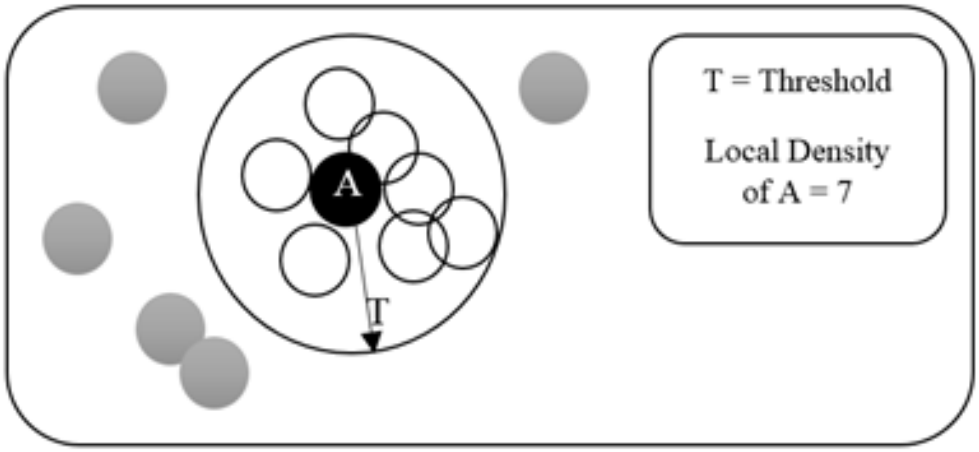


Figure 3: The local density of a document

δ_i is the similarity between document i and the most similar document with higher local density. For example, as we can see in Figure 4, δ for the black document is the similarity between the gray document and itself.

Eventually, we suggest outliers that are more different from the behavior of central data. In other words, outliers are texts with less ρ_i and fewer δ_i.

Considering the purpose of finding outliers in text data and density peak, the rest of this section explains our new method. First of all, the similarity of each pair of texts is calculated and saved in a matrix (See Figure 5).

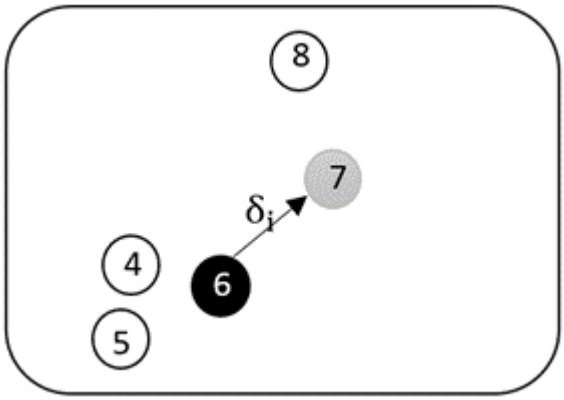


Figure 4: The computing of δ_i

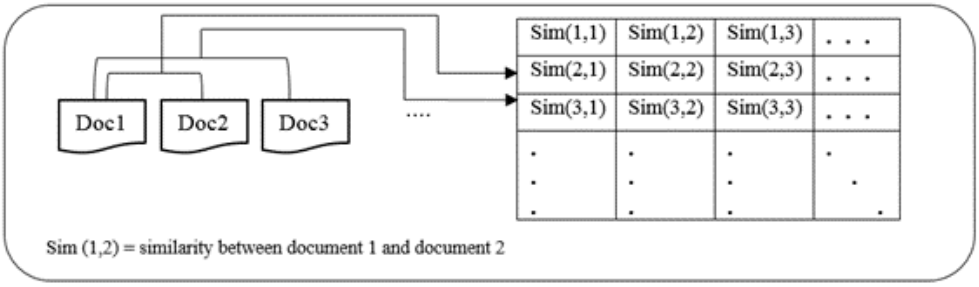


Figure 5: The computing of similarity matrix

Then a threshold should be set for the similarity between documents. Now for each document, ρ_i is computed and saved in the ρ array. Afterward, δ_i should be recognized for all documents and saved in the δ array. Finally, outliers are searched as the ρ_i and δ_i are lower than the thresholds. We can see the steps of the algorithm in Algorithm 1.

Algorithm 1
Input: documents, $TS, Td, TS1$
Output: Outliers
1. Calculate the similarity index
2. Set the ρ array
3. Set the δ array
4. Print the number of documents in which the $\rho_i < T_d$ and $\delta_i < TS1$ as outliers

3.1 Dataset

BBC dataset is utilised in our experiment to evaluate the performance of our suggested method for outlier spotting in the text. Different scenarios are used to test each dataset. First, several outlier identification methods prepare them for entrance into categorization algorithms. In these situations, data identified using our suggested approach, LOF, and DBSCAN is cleaned; otherwise, the datasets remain untouched. The accuracy of four distinct categorization methods is then calculated, and they are contrasted. These tests are designed to find out how our strategy of eliminating outliers affects classification algorithm performance.

3.2 Classification algorithms

In our experiments, four different classification algorithms (K Nearest Neighbor, Decision Tree, Random Forest, and Naïve Bayes) are implemented to compare different situations. Furthermore, Term Frequency (TF) [27], Term Frequency-Inverse Document Frequency(TF-IDF) [36], and 3-gram [9] are used for the inputs of these algorithms. We briefly describe these four algorithms in the rest of this section.

3.2.1 K-nearest neighbor (KNN)

The k nearest neighbor classifier is a classification algorithm based on the distance between the input sample and the training samples. The algorithm will find the k objects which are closest to the input data, and the value of the input will be predicted according to the values of these neighbors. For instance, xi is input, and $(x_{i_1}, xi_2, ..., x_{i_p})$ are its features. The Euclidean distance between x_i and x_l is computed as:

$$d(x_i, x_l) = \sqrt{(x_{i_1} - x_{l_1})^2 + (x_{i_2} - x_{l_2})^2 + ... + (x_{i_2} - x_{l_2})^2} \tag{5}$$

This distance is computed between x_i and every training sample to find the nearest neighbors and the value of x_i is considered regarding its K nearest neighbors [19, 29], as shown in Figure 6.

3.2.2 Decision Tree (DT)

One of the flowchart-like structure methods is a decision tree. It makes a tree based on training data when new data enters into the training model which is similar to the yes and no question game, predicts the input value. Each tree

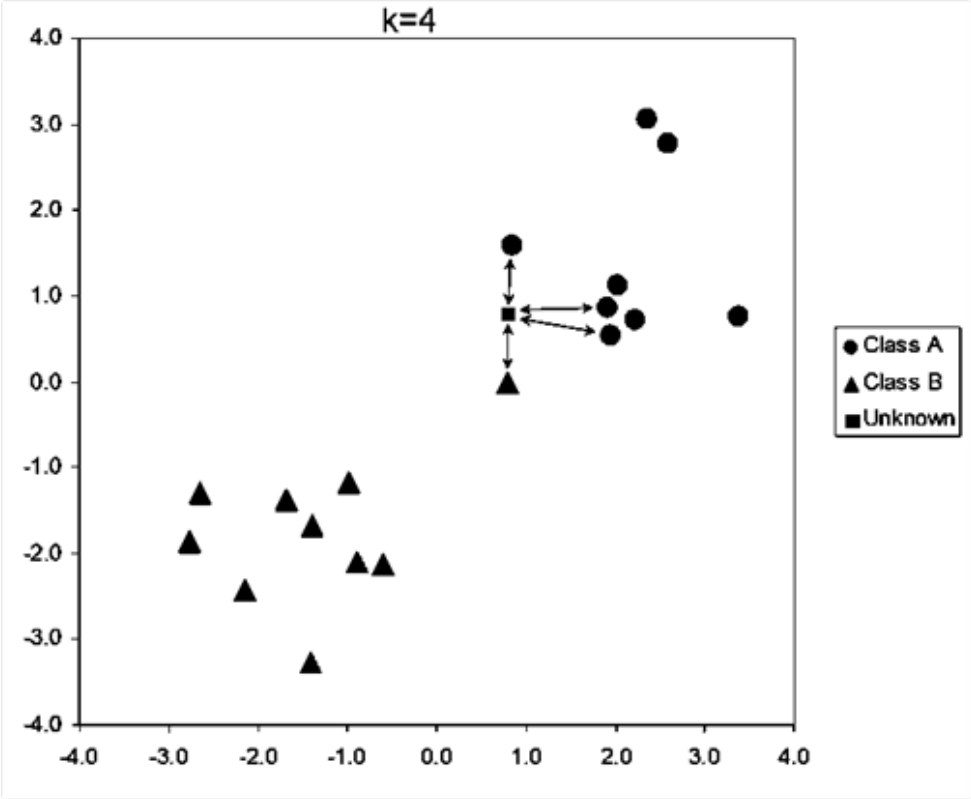


Figure 6: The vote for the unknown sample values recognition [26]

composes of nodes and branches. The nodes show features of classes, and they can get values by using branches. Figure 7 represents the decision tree as an example. Decision trees have found many fields of implementation due to their simple analysis [32].

3.2.3 Random Forest (RF)

Random Forest is an ensemble learning method for classification, regression, and other learning that builds a large number of decision trees during training. This algorithm makes a number of random decision trees with different properties, then the value of new data is decided by the voting of these trees [15]. Figure 8 illustrates the logic of the random forest algorithm.

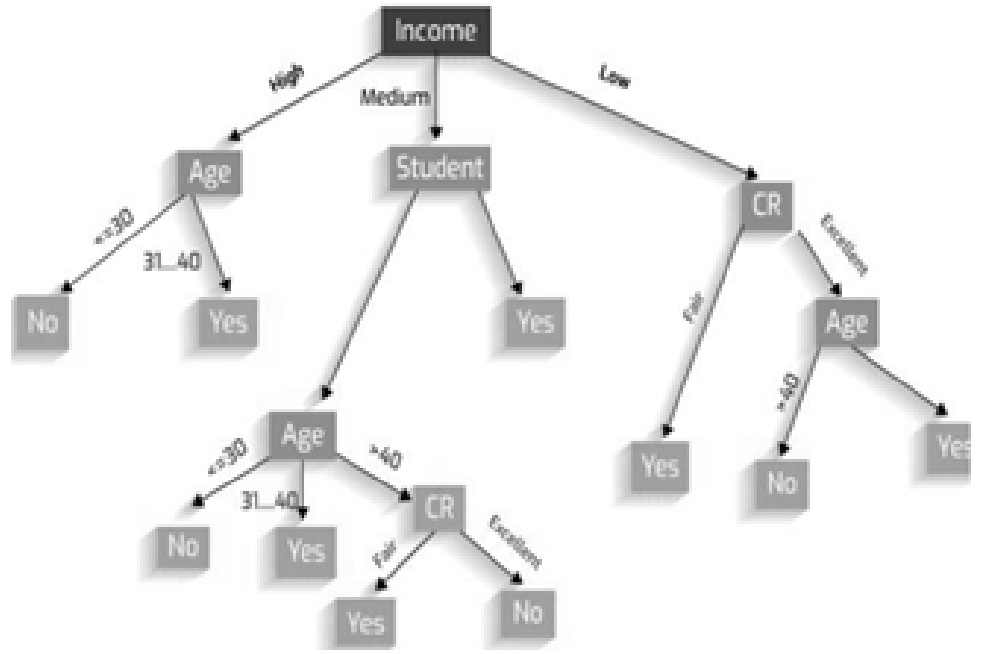


Figure 7: The example of a decision tree [7]

3.2.4 Naïve Bayes

Despite the simple design and assumptions, Naïve Bayes classifiers have worked very well in many complex real-world situations. Naïve Bayes is a probability model based on Bayes' theorem which is defined as:

$$P(A|B) = \frac{p(B|A).p(A)}{p(B)} \tag{6}$$

This algorithm calculates the probability of each input feature in each class by considering the training data and selects the most likely class as the value of the input sample. In this method, X is the input, and $(x_1, x_2, \ldots, x_i)$ are its features. The values or classes are defined as $(C_1, C_2, \ldots, C_i)$. Now for each input, the probability of each class for the input is predicted by Equation 7 [14].

$$P(C_i|X) = \frac{p(X|C_i).p(Ci)}{p(X)} \tag{7}$$

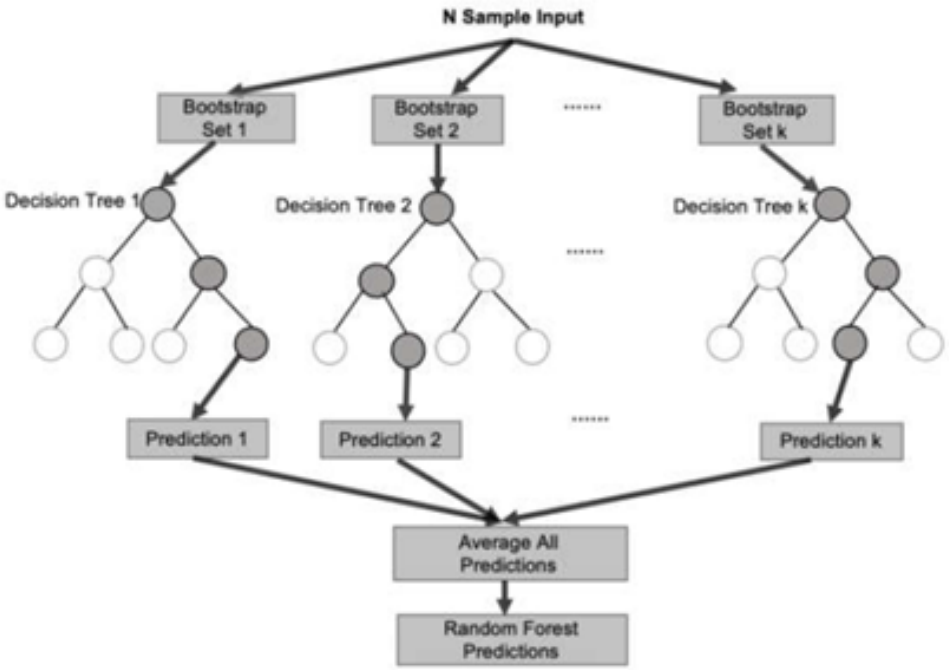


Figure 8: The process of decision in the Random Forest [39]

That $p(X|C_i)$ is calculated as:

$$P(X|C_i) = p(X|C_i) = \prod_{k=1}^{n} p(x_k|C_i) = P(x_1|C_i).P(X_2|C_i)...P(x_n|C_i) \quad (8)$$

4 Experimental results

According to Table 1, the proposed model was implemented using the Python programming language. To implement the models, two parts of the dataset were used for model training and evaluation. There is the 80% training dataset and the 20% evaluation dataset.

OS	Windows 10-64bit
CPU	Intel(R) Core (TM)i7-7700HQ 2.8GHz
GPU	NVIDIA Tesla P100
RAM	16 GB
Programming language	Python 3.9
Software libraries	TensorFlow, Keras, OpenCV, and Scikit-Learn

Table 1: Hardware & software environments deployed in this study

4.1 Evaluation metrics for classification problems

Evaluation of any machine learning model's performance is the most crucial step in model construction. So, the question of how to evaluate a machine learning model's performance arises. Machine learning tasks are connected to evaluation measures. Regression and classification tasks each have their own metrics. Before we put our model into production on untested data, we should be able to increase its overall predictive power by evaluating its performance using several criteria. When a machine learning model is applied to unexplored data, failing to properly evaluate it using a variety of assessment measures and relying simply on accuracy can result in inaccurate predictions. Accuracy, confusion matrix, precision, recall, F1-score, sensitivity, specificity, and AUC are major performance measures for classification problems that are most frequently employed. Following equations represent these metrics:

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{9}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{10}$$

$$\mathrm{F1\text{-}score} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{11}$$

$$\mathrm{Sensitivity} = \mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{12}$$

$$\mathrm{Specificity} = \frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}} \tag{13}$$

Instances where the model accurately predicts a positive class are referred to as TPs (True Positives). This is taken to be true since the input really corresponds

to the positive class that the model predicted would exist. FP (False Positive) is the term for when a positive class is falsely predicted by a model, even if it can be perceived as the model doing so. FN (False Negative) is the term used when a model predicts a negative class wrongly, which is untrue but can be read as a negative class being predicted by the model. When the model properly predicts a negative class, this is known as a TN (True Negative), and is taken to be true because the input really corresponds to the predicted negative class.

Classification accuracy is the most often used performance metric for evaluating classification models. Due to the fact that it can be stated as a single number that encapsulates the model's capabilities and is straightforward to compute and comprehend, it is widely used. Therefore, in our simulations, accuracy was the performance indicator we used.

4.2 Simulation results

As can be seen in Table 2, all algorithms, including Decision Tree, Random Forest, and Naïve Bayes, performed better once outliers were removed using our method. Therefore, our approach significantly impacts when used as a pre-processing algorithm during text processing. Results demonstrate that our strategy improves the accuracy of the Nave Bayes classifier by more than 98%.

Table 3 compares the accuracy of our method with DBSCAN, and LOF for Random Forest algorithm. It constructs a forest using a collection of decision trees. Random Forests produce uncorrelated decision trees and execute feature selection implicitly. To accomplish, it constructs each decision tree using a random collection of features. This makes it a great model for working with data that has a lot of different properties. The results show that when Random Forests were used, our method's accuracy increased from 90% to 92.5%, despite the fact that Random Forests are not much influenced by outliers. This highlights how effective our approach is at detecting outliers.

The performance of our suggested detection of outliers technique will then be assessed when used for The K-Nearest Neighbor (KNN). The KNN machine learning technique is flexible. It is employed in a variety of contexts, including handwriting recognition, picture recognition, and video recognition. In a wide range of prediction problems, it can achieve high accuracy. KNN is a method that is used to learn an unknown function with the appropriate precision, and accuracy. It is based on the local minimum of the target function. The algorithm also determines a parameter's range or distance from an unknown input as well as its surroundings. Based on the "information gain" theory, the

Algorithm	Decision Tree		
Outlier detection		Without deleting outliers	Our method
Accuracy	TF	0.8071	0.8669
	TF_IDF	0.8071	0.8394
	N-gram (n=3)	0.7533	0.7114
Algorithm	Random Forest		
Outlier detection		Without deleting outliers	Our method
Accuracy	TF	0.8917	0.9149
	TF_IDF	0.9058	0.9278
	N-gram (n=3)	0.8466	0.8522
Algorithm	Naïve Bayes		
Outlier detection		Without deleting outliers	Our method
Accuracy	TF	0.9596	0.9862
	TF_IDF	0.9686	0.9816
	N-gram (n=3)	0.9327	0.9633

Table 2: The accuracy of Decision Tree, Random Forest, and Naïve Bayes algorithms on the BBC dataset

Technique	With outliers	LOF (k=7)	DBSCAN (3,2)	Proposed
TF	0.8917	0.9197	0.9548	0.9149
TF IDF	0.9058	0.9217	0.9031	0.9278
N-gram	0.8466	0.8235	0.8463	0.8522

Table 3: Comparing the accuracy of our method with DBSCAN, and LOF for Random Forest algorithm

algorithm determines which method is best suited to forecast an unknowable value. Figures 9, 10, and 11 demonstrate that our method along with KNN can get an accuracy more than 96% for K=10 and TF-IDF. It strongly outperforms other existing approaches presented in these figures in particular when an outlier detector was not implemented.

Using a k-fold cross-validation approach, we expanded our experimental findings and verified the effectiveness of our model. Our dataset was separated into k subsets ($k = 5$), the model was trained on k-1 subsets, and it was then assessed on the final subset. A new subset was tested in each of the k iterations of this process which was repeated. The performance of the model was then estimated by averaging the findings. The outcomes of our tests revealed that,

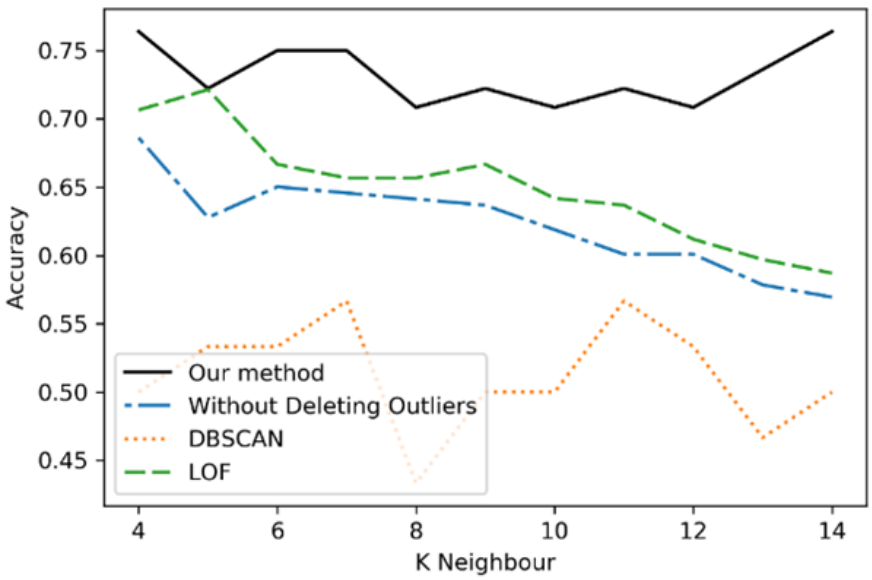


Figure 9: The accuracy of KNN using TF and k between 4 and 14 on the BBC dataset

when tested using cross-validation, our model had somewhat enhanced the performance to reach 98.92% accuracy.

5 Conclusion

Today, many applications leverage data mining and machine learning approaches ([1, 4, 11]). Given the fierce competition to obtain ever-increasing accuracy, any beneficial pre-processing procedures, such as outlier detection, would be taken into consideration in real systems. Additionally, text features indicate that processing this kind of data is challenging. Nevertheless, processing can be made smoother by spotting anomalous data. In this research, we have presented a unique method for identifying anomalous data in text datasets. The characteristics of the texts that we uncovered did not match those in the clusters' center. We created the density-based method in keeping with this idea. Then, we conducted a number of experiments to evaluate how well our approach performed compared to two popular outlier detectors

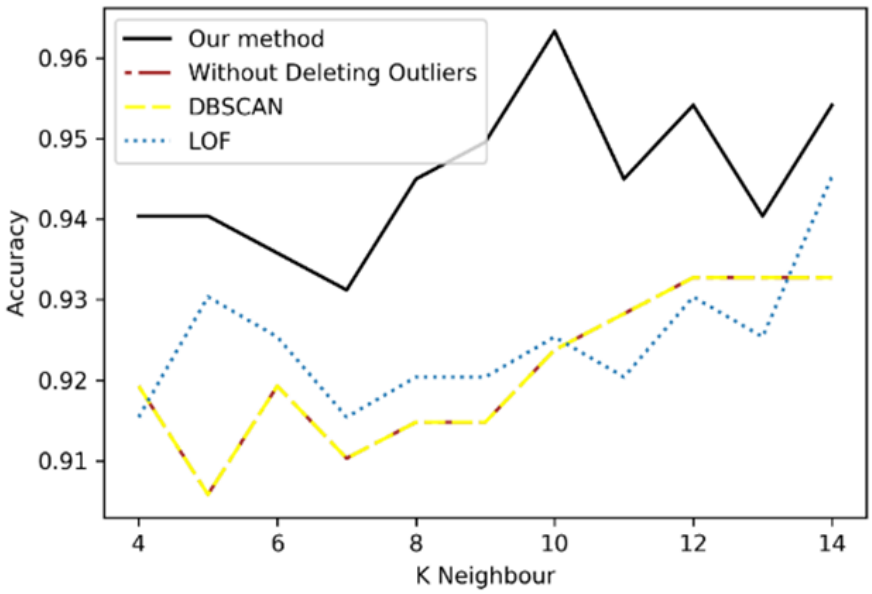


Figure 10: The accuracy of KNN on the BBC dataset with k between 4 to 14 with TF-IDF

(LOF and DBSCAN). For our tests, we used the BBC dataset. In the majority of cases, our recommended approach outperformed LOF and DBSCAN techniques. After pre-processing with our technique, the accuracy is shown to rise and the KNN algorithm performs better overall. It maintains first place with a significant disparity.

Ultimately, our proposed approach can be used for both short and lengthy texts and can be applied to most datasets without taking into account the system knowledge. Ultimately, the versatility of our approach may be increased by making use of additional similarity methods, such semantic similarity.

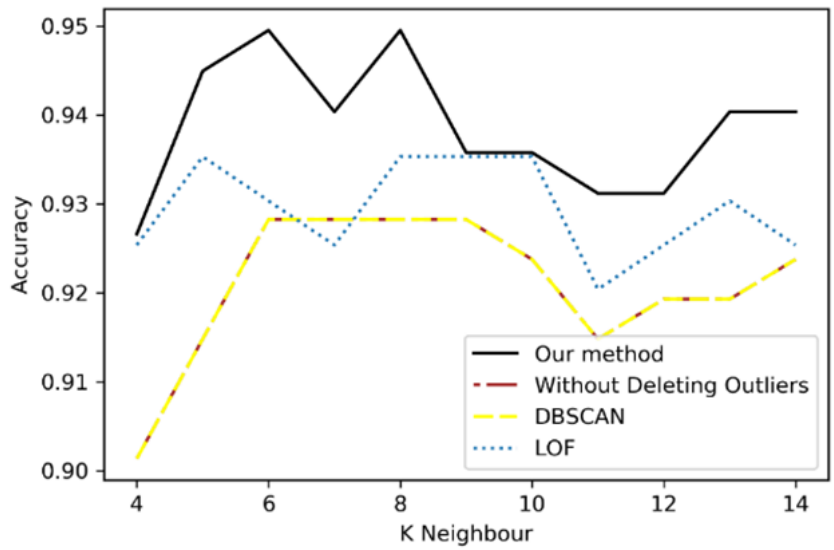


Figure 11: The accuracy of KNN on the BBC dataset with k between 4 to 14 with N-gram vectors

References

[1] F. Abedini, M. Bahaghighat, M. S'hoyan, Wind turbine tower detection using feature descriptors and deep learning. *Facta Universitatis, Series: Electronics and Energetics*, **33,** 1 (2019) 133–153. ⇒105

[2] J. Allan, V. Lavrenko, D. Malin, R. Swan, Detections, bounds, and timelines: Umass and tdt-3. In *Proceedings of Topic Detection and Tracking Workshop*, pp. 167–174. Citeseer, 2000. ⇒92

[3] M. Bahaghighat, F. Abedini, Q: Xin, M. Mohammadi Zanjireh, S. Mirjalili, Using machine learning and computer vision to estimate the angular velocity of wind turbines in smart grids remotely. *Energy Reports*, **7** (2021) 8561–8576. ⇒92

[4] M. Bahaghighat, Q. Xin, S. Ahmad Motamedi, M. Mohammadi Zanjireh, A. Vacavant, Estimation of wind turbine angular velocity remotely found on video mining and convolutional neural network. *Applied Sciences*, **10,** 10 (2020) 3544. ⇒105

[5] C. Barreyre, L. Boussouf, B. Cabon, B. Laurent, J-M. Loubes, Statistical methods for outlier detection in space telemetries. *Space Operations: Inspiring Humankind's Future*, pp. 513–547, 2019. ⇒93

[6] I. Ben-Gal, *Outlier detection in: Data mining and knowledge discovery handbook: A complete guide for practitioners and researchers*, 2005. ⇒93

[7] Y. Bengio, O. Delalleau, C. Simard, Decision trees do not generalize to new variations. *Computational Intelligence*, **26,** 4 (2010) 449–467. ⇒100

[8] M. Bozorgi, M. Mohammadi Zanjireh, M. Bahaghighat, Q. Xin, A time-efficient and exploratory algorithm for the rectangle packing problem. *Intelligent Automation & Soft Computing*, **31,** 2 (2022) 885–898. ⇒92

[9] A. Z. Broder, S. C. Glassman, M. S Manasse, G. Zweig, Syntactic clustering of the web. *Computer networks and ISDN systems*, **29,** 8–13 (í997) 1157–1166. ⇒98

[10] M. Ester, H-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, vol. 96, pp. 226–231, 1996. ⇒93

[11] M. Ghorbani, M. Bahaghighat, Q. Xin, F. Özen, ConvLSTMconv network: a deep learning approach for sentiment analysis in cloud computing. *Journal of Cloud Computing*, **9,** Article no: 16 (2020). ⇒92, 105

[12] J. Guzman, B. Poblete, On-line relevant anomaly detection in the twitter stream: an efficient bursty keyword detection model. In *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, pp. 31–39, 2013. ⇒92, 94

[13] A. Hajikarimi, M. Bahaghighat, Optimum outlier detection in internet of things industries using autoencoder. In *Frontiers in Nature-Inspired Industrial Optimization*, pp. 77–92, 2022. ⇒92

[14] D. J. Higham, An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review*, **43,** 3 (2001) 525–546. ⇒100

[15] T. K. Ho, Random decision forests. In *Proc. of 3rd Int. Conf. on Document Analysis and Recognition*, vol. 1. pp. 278–282. IEEE, 1995 ⇒99

[16] V. Hodge, J. Austin, A survey of outlier detection methodologies. *Artificial Intelligence Review*, **22** (2004) 85–126. ⇒92

[17] M. Jamalzadeh, M. Maadani, M. Mahdavi, Ec-mopso: an edge computing-assisted hybrid cluster and mopso-based routing protocol for the internet of vehicles. *Annals of Telecommunications*, **77,** 7–8 (2022) 491–503. ⇒93

[18] S. M. Jameii, M. Maadani, Intelligent dynamic connectivity control algorithm for cluster-based wireless sensor networks. In *2016 11th Int. Conf. for Internet Technology and Secured Transactions (ICITST)*, pp. 416–420. IEEE, 2016. ⇒93

[19] T. Joachims, *A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization.* Technical Report, Carnegie-Mellon Univ. Pittsburgh. Dept. of Computer Science, 1996. ⇒98

[20] S. Kannan, V. Gurusamy, S. Vijayarani, J. Ilamathi, Ms. Nithya, S. Kannan, V. Gurusamy, Preprocessing techniques for text mining. *International Journal of Computer Science & Communication Networks*, **5,** 1 (2014) 7–16. ⇒92

[21] F. Khorasani, M. Mohammadi Zanjireh, M. Bahaghighat, Q. Xin, A tradeoff between accuracy and speed for k-means seed determination. *Comput. Syst. Sci. Eng.*, **40,** 3 (2022) 1085–1098. ⇒92

[22] B. S. Kumar, V. Ravi, A survey of the applications of text mining in financial domain. *Knowledge-Based Systems*, **114** (2016) 128–147. ⇒92

[23] R. Kumaraswamy, A. Wazalwar, T. Khot, J. Shavlik, S. Natarajan, Anomaly detection in text: The value of domain knowledge. In *The Twenty-Eighth International Flairs Conference*, 2015. ⇒92

[24] Y. Li, Z. Chen, D. Zha, K. Zhou, H. Jin, H. Chen, X. Hu. Autood: Automated outlier detection via curiosity-guided search and self-imitation learning. *arXiv preprint arXiv:2006.11321*, 2020. ⇒92

[25] Y. Liu, Z. Li, Ch. Zhou, Y. Jiang, J. Sun, M. Wang, X. He, Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, **32,** 8 (2019) 1517–1528. ⇒93

[26] A. R. Lubis, M. Lubis, et al., Optimization of distance formula in k-nearest neighbor method. *Bulletin of Electrical Engineering and Informatics*, **9,** 1 (2020) 326–338. ⇒99

[27] H. P. Luhn, A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, **1,** 4 (1957) 309–317. ⇒98

[28] M. Norouzi Shad, M. Maadani, M. Nesari Moghadam, Gapso-Svm: an IDSS-based energy-aware clustering routing algorithm for IoT perception layer. *Wireless Personal Communications*, **216** (2022) 2249–2268. ⇒93

[29] M. Oghbaie, M. Mohammadi Zanjireh, Pairwise document similarity measure based on present term set. *Journal of Big Data*, **5,** 1 (2018) 1–23. ⇒98

[30] M. Platakis, D. Kotsakos, D. Gunopulos, Searching for events in the blogosphere. In *Proceedings of the 18th Int. Conf. on World Wide Web*, pp. 1225–1226, 2009. ⇒92

[31] X. Qin, L. Cao, E. A. Rundensteiner, S. Madden, Scalable kernel density estimation-based local outlier detection over large data streams. In *Proceedings of the 22nd Int. Conf. on Extending Database Technology (EDBT)*, 2019. ⇒93

[32] J. P. Reiter, T. E. Raghunathan, The multiple adaptations of multiple imputation. *Journal of the American Statistical Association*, **102,** 480 (2007) 1462–1471. ⇒99

[33] M. Rostami, M. Bahaghighat, M. Mohammadi Zanjireh, Bitcoin daily close price prediction using optimized grid search method. *Acta Universitatis Sapientiae, Informatica*, **13,** 2 (2021) 265–287. ⇒92

[34] S. N. Sajedi, M. Maadani, M. Nesari Moghadam, F-leach: a fuzzy-based data aggregation scheme for healthcare IoT systems. *The Journal of Supercomputing*, **78,** 1 (2022) 1030–1047. ⇒92

[35] E. Schubert, M. Weiler, H-P. Kriegel, Signitrend: scalable detection of emerging topics in textual streams by hashed significance thresholds. In *Proceedings of the 20th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 871–880, 2014. ⇒92

[36] H. Schütze, Ch. D. Manning, P. Raghavan, *Introduction to information retrieval*, vol. 39. Cambridge University Press Cambridge, 2008. ⇒98

[37] A. Shamseen, M. Mohammadi Zanjireh, M. Bahaghighat, Q. Xin, Developing a parallel classifier for mining in big data sets. *IIUM Engineering Journal*, 22, 2 (2021) 119–134. ⇒92, 95

[38] M: Templ, J. Gussenbauer, P. Filzmoser, Evaluation of robust outlier detection methods for zero-inflated complex data. *Journal of Applied Statistics*, 47, 7 (2020) 1144–11673. ⇒92

[39] B. Wang, J. Sharma, J. Chen, P. Persaud, Ensemble machine learning assisted reservoir characterization using field production data–an offshore field case study. *Energies*, 14, 4 (2021) 1052. ⇒101

[40] Y. Wu, X. Li, F. Luan, Y. He, A novel gpr-based prediction model for strip crown in hot rolling by using the improved local outlier factor. *IEEE Access*, **9** (2020) 458–469. ⇒94

[41] Y. Yan, L. Cao, C. Kulhman, E. Rundensteiner, Distributed local outlier detection in big data. In *Proceedings of the 23rd ACM SIGKDD Int. Conference on knowledge Discovery and Data Mining*, pp. 1225–1234, 2017. ⇒92, 93

[42] Y. Zhao, Z. Nasrullah, Z. Li, PyOD: A Python toolbox for scalable outlier detection. *arXiv preprint arXiv:1901.01588*, 2019. ⇒92

Received: May 5, 2023 • *Revised: June 21, 2023*

Acta Univ. Sapientiae Informatica **15,** 1 (2023) 111–123

DOI: 10.2478/ausi-2023-0009

Eccentric connectivity index in transformation graph G^{xy+}

Aysun AYTAÇ
Faculty of Sciences
Department of Mathematics
Ege University
İzmir , Turkey
email: aysun.aytac@ege.edu.tr

Belgin VATANSEVER
Faculty of Sciences
Department of Mathematics
Ege University
İzmir, Turkey
email: belgin.vatansever@ege.edu.tr

Abstract. Let G be a connected graph with vertex set $V(G)$and edge set $E(G)$. The eccentric connectivity index of G is defined as $\sum_{v \in V(G)} ec(v)\deg(v)$ where $ec(v)$ the eccentricity of a vertex v and $\deg(v)$is its degree and denoted by $\varepsilon^c(G)$. In this paper, we investigate the eccentric connectivity index of the transformation graph G^{xy+}.

1 Introduction

A topological index is a number that describes a molecular structure and is obtained from the associated (hydrogen-depleted) molecular graph. Topological indices are mathematical properties of graphs that are utilized to establish relationships between the structural properties of chemical molecules and their physical attributes. The aforementioned indices are extensively utilized in the fields of quantitative structure-activity relationship (QSAR) and quantitative structure-property relationship (QSPR), chemical documentation and drug design studies [6, 7, 8, 11, 13, 15].

Key words and phrases: Distance, eccentricity, eccentric connectivity index, transformation graph.

In pharmaceutical research, QSAR data is utilized to identify the most viable compounds with respect to a specific property, thereby reducing the number of compounds that must be synthesized in the process of designing new drugs. Despite the fact that many topological indices have been described, only a small number of them have been used effectively in QSAR investigations. These include Wiener's index, Balaban's index, Hosoya's index, Randic's molecular connectivity index, and the eccentric connectivity index [2, 10, 9, 21]. Eccentricity has been used to create a variety of indexes [3, 12, 4, 5, 14, 20] . Some of these are eccentric connectivity index, graph shape index, and connective eccentricity index. In this study, we discussed the index which is defined, in 1997, by Sharma et al., as eccentric connectivity [14]. The eccentric connectivity index $\varepsilon^c(G)$ of G is defined as $\varepsilon^c(G) = \sum_{v \in V(G)} ec(v)\deg(v)$.

Consider a simple connected graph denoted by G with its set of vertices represented as $V(G)$ and the set of edges as $E(G)$. The metric that quantifies the distance between two vertices u and v in a graph G, denoted by $d_G(u,v)$, is defined as the minimum number of edges that must be traversed in order to travel from u to v along the shortest path in G. The vertex eccentricity, denoted as $ec_G(u)$, in a graph G refers to the greatest distance between vertex u and any other vertex in G. The mathematical definition of the diameter, denoted as d, of a graph G is the largest possible value of the eccentricities of all vertices in G. The definition of the radius of a graph G is such that it corresponds to the minimum value of the eccentricities of the vertices that comprise G. In graph theory, a vertex in a graph G is considered to be central if its eccentricity is equivalent to the radius of G. The number of edges that are connected to a vertex $w \in V(G)$ is defined as the degree of the vertex, denoted by $\deg_G(w)$. A graph theory term for a vertex with only one adjacent vertex is a pendant vertex, also known as a leaf vertex, of a given graph G. The open neighborhood and closed neighborhood of a vertex v in a graph G are defined as $N_G(v) = \{u \in V(G) : uv \in E(G)\}$ and $N_G[v] = N_G(v) \cup \{v\}$, respectively. Let the set $N_G^i(v)$ be the set of vertices where the vertex v is at a distance i in the graph G. That is, $N_G^i(v) = \{u \in V(G) \mid d(v,u) = i\}$. Thus, we have $N(v) = N_G(v) = N_G^1(v)$ and $N[v] = N_G[v] = N_G^1(v) \cup \{v\}$ [16].

The vertex set of the complement $\bar{G}$ of a graph G consists of the same vertices as G, but in $\bar{G}$, two vertices are adjacent if and only if they are not adjacent in G. On the other hand, the line graph $L(G)$ of G is a graph whose vertex set is composed of the edges of G, and two vertices in $L(G)$ are adjacent if and only if the corresponding edges are adjacent in G [16].

The transformation graph G^{xyz} is a graph whose vertex set is $V(G) \cup E(G)$, and $s, t \in V(G^{xyz})$. The vertices s and t are adjacent in G^{xyz} if and only if one of the following properties holds [1, 19, 18, 17]:

(P1) Consider $s, t \in V(G)$. If $x = +$, then $t \in N_G(s)$; while if $x = -$, then $t \notin N_G(s)$.
(P2) Consider $s, t \in E(G)$. If $y = +$, then $t \in N_G(s)$; while if $y = -$, then $t \notin N_G(s)$.
(P3) Consider $s \in V(G), t \in E(G)$. If $z = +$, then s is the end-vertex of t; while if $z = -$, then s is not the end-vertex of t.

In this paper, we study about eccentric connectivity index of the transformation graph G^{xy+}. Various notations are employed to enhance the comprehensibility of the proofs of the aforementioned theorems. Consider two arbitrary vertices s and t in the graph G. In the context of graph theory, it is customary to denote the edge between two adjacent vertices s and t in a graph G as e_{st}. Moreover, the aforementioned edge is denoted by the vertex st within the graph G^{xyz}.

Theorem 1 [22] *Let G be a connected graph with m edges. Then,*

$$2m(\mathrm{rad}(G)) \leq \varepsilon^c(G) \leq 2m(\mathrm{diam}(G)).$$

2 Eccentric connectivity index for the graph G^{xy+}

We begin this subsection by determining the eccentric connectivity index of the transformation graph G^{xy+} when G is a specified family of graphs.

Theorem 2 *When $xyz = +-+$, let the transformation graph of the graph G be G^{+-+} and q is the number of edges of the graph G^{+-+}.*

(a) If $G \cong P_n$ $(n \geq 6)$, then $\varepsilon^c(G^{+-+}) = 2n^2 + 6n - 4$;
(b) If $G \cong C_n$ $(n \geq 6)$, then $\varepsilon^c(G^{+-+}) = 2n^2 + 10n$;
(c) If $G \cong K_n$ $(n \geq 4)$, then $\varepsilon^c(G^{+-+}) = (n-1)\left(\frac{n^3-5n^2+18n}{2}\right) = 4q$;
(d) If $G \cong K_{1,n}$ $(n \geq 3)$, then $\varepsilon^c(G^{+-+}) = 10n$;
(e) If $G \cong W_{1,n}$ $(n \geq 3)$, then $\varepsilon^c(G^{+-+}) = 6n^2 + 10n = 4q$;
(f) If $G \cong K_{m,n}$ $(m, n \geq 2)$, then $\varepsilon^c(G^{+-+}) = 2mn(mn - m - n + 7)$;

Proof. (a) Let x and y be the pendant vertices of the graph G. It is easily seen that $\deg_{G^{+-+}}(x) = \deg_{G^{+-+}}(y) = 2$ and for all $v \in V(G) - \{x, y\}$ $\deg_{G^{+-+}}(v) = 4$. Furthermore, $\deg_{G^{+-+}}(xN_G(x)) = \deg_{G^{+-+}}(yN_G(y)) = n - 1$ and for $uv \in V(\overline{L(G)}) - \{xN_G(x),\ yN_G(y))$ $\deg_{G^{+-+}}(uv) = n - 2$. The eccentricity value of the vertices of the graph is calculated according to the vertices as follows.

- $e(u)$ value of for all $u \in V(G)$:
  Let $A = N_G(u),\ B = V(G) - N_G[u],\ C = uN_G(u),\ D = V(\overline{L(G)}) - C$. The shortest distance between the vertex u and the vertices in $A \cup C$, in $D \cup N_G^2(u)$ and in $V(G) - (N_G[u] \cup N_G^2(u))$ is 1, 2 and 3, respectively. Thus, we get $e(u) = 3$.

- $e(uv)$ value of for all $uv \in V(\overline{L(G)})$:
  Let $A = N_{\overline{L(G)}}(uv)$ and $B = V(L(G)) - N_{\overline{L(G)}}[uv]$. The shortest distance between the vertex uv and the vertices in $A \cup \{u, v\}$, in $N_G(u) - \{v\} \cup N_G(v) - \{u\} \cup N_G(A)$ and in $B = N_{\overline{L(G)}}(A) - N_{\overline{L(G)}}[uv]$ is 1, 2 and 2, respectively. Thus, we get $e(uv) = 2$.
  With the results we found, we get, for $n \geq 6$,

$$\begin{aligned}\varepsilon^c\left(G^{+-+}\right) &= \sum \deg(v)e(v) \\ &= 2 \cdot 2 \cdot 3 + (n-2) \cdot 4 \cdot 3 + 2 \cdot (n-1) \cdot 2 + (n-3)(n-2) \cdot 2 \\ &= 2n^2 + 6n - 4.\end{aligned}$$

(b) We can easily observe that for $\forall v \in V(G)$ $\deg_{G^{++}}(v) = 4$ and for $uv \in V(\overline{L(G)})$ $\deg_{G^{+}}(uv) = n - 1$. The eccentricity value of the vertices of the graph is calculated according to the vertices as follows.

- $e(u)$ value of $\forall u \in V(G)$: Let $A = N_G(u), B = V(G) - N_G[u], C = uN_G(u)$ and $D = V(\overline{L(G)}) - C$. The shortest distance between the vertex u and the vertices in $A \cup C$, in $D \cup N_G^2(u)$ and in $V(G) - \left(N_G[u] \cup N_G^2(u)\right)$ is 1, 2 and 3, respectively. Thus, we get $e(u) = 3$.

- $e(uv)$ value of $\forall uv \in V(\overline{L(G)})$: Let $A = N_{\overline{L(G)}}(uv)$ and $B = V(\overline{L(G)}) - N_{\overline{L(G)}}[uv]$. The shortest distance between the vertex uv and the vertices in $A \cup \{u, v\}$, in $N_G(u) - \{v\} \cup N_G(v) - \{u\} \cup N_G(A)$ and in $B = N_{\overline{L(G)}}(A) - N_{\overline{L(G)}}[uv]$ is 1, 2 and 2, respectively. Thus, we get $e(uv) = 2$.

  With the results we found, we get, for $n \geq 6$,

  $\varepsilon^c\left(G^{+-+}\right) = \sum \deg(v)e(v) = n \cdot 4 \cdot 3 + n \cdot (n-1) \cdot 2 = n^2 + 10n.$

(c) It can easily be observed that for $\forall v \in V(G)\, deg_{G^{+-+}}(v) = 2(n-1)$ and for $uv \in V(\overline{L(G)})\, deg_{G^{+-+}}(uv) = \frac{n^2-5n+10}{2}$. The eccentricity value of the vertices of the graph is calculated according to the vertices as follows.

- $e(u)$ value of $\forall u \in V(G)$: Let $A = N_G(u), B = V(G) - N_G[u] = \varnothing, C = uN_G(u)$ and $D = V(\overline{L(G)}) - C$. The shortest distance between the vertex u and the vertices in $A \cup C$ and in $D = N_{\overline{L(G)}}(C)$ is 1 and 2, respectively. Thus, we get $e(u) = 2$.

- $e(uv)$ value of $\forall uv \in V(\overline{L(G)})$: Let $A = N_{\overline{L(G)}}(uv), B = V(\overline{L(G)}) - N_{\overline{L(G)}}[uv], C = N_G(uv)$ and $D = V(G) - C$. The shortest distance between the vertex uv and the vertices in $A \cup C$, in $B = N_{\overline{L(G)}}(A) - N_{\overline{L(G)}}[uv]$ and D is 1, 2 and 1, respectively. Thus, we get $e(uv) = 2$. With the results we found, we get, for $n \geq 4$,

$$\varepsilon^c\left(G^{+-+}\right) = \sum deg(v)e(v) = n.2(n-1).2 + \frac{n(n-1)}{2}.\frac{n^2-5n+10}{2}.2$$
$$= (n-1)\left(\frac{n^3-5n^2+18n}{2}\right)$$

Furthermore, since the graph's vertices have an eccentricity value of 2, according to Theorem 1, $\varepsilon^c\left(G^{+-+}\right) = 4q$ where q is the number of edges of the graph G^{+-+}.

(d) Let c be the central vertex of the graph G. It is easily seen that $deg_{G^{+-+}}(c) = 2n$ and for $\forall v \in V(G) - \{c\}$ $deg_{G^{+-+}}(v) = 2$. Since the structure of $\overline{L\left(K_{1,n}\right)}$ consists of n isolated vertices, $deg_{G^{+-+}}\left(cN_G(c)\right) = 2$. Also, $e(c) = 1$ and for $\forall w \in V\left(G^{+-+}\right) - \{c\}, e(w) = 2$.

With the results we found, we get, for $n \geq 3$,

$\varepsilon^c\left(G^{+-+}\right) = \sum deg(v)e(v) = 2 \cdot n + 2 \cdot n \cdot 2 \cdot 2 = 10n.$

(e) Let c be the central vertex of the graph G. It is easy to see that $deg_{G^{+-+}}(c) = 2n$ and for $\forall v \in V(G) - \{c\}\, deg_{G^{+-+}}(v) = 6$. For the vertices corresponding to the edges connecting the central vertex and the vertices on the cycle graph, $deg_{G^{+-+}}\left(cN_G(c)\right) = n$ and for $\forall uv \in V(\overline{L(G)}) - \{cN_G(c)\}\, deg_{G^+}(uv) = 2n-3$. The graph's vertices have an eccentricity value of 2.

With the results we found, we get, for $n \geq 3$,

$$\varepsilon^c\left(G^{+-+}\right) = \sum deg(v)e(v) = 1{\cdot}2n{\cdot}2{+}n{\cdot}6{\cdot}2{+}n{\cdot}n{\cdot}2{+}n{\cdot}(2n{-}3){\cdot}2 = 6n^2{+}10n.$$

Furthermore, since the graph's vertices have an eccentricity value of 2, according to Theorem 1, $\varepsilon^c(G^{+-+}) = 4q$ where q is the number of edges of the graph G^{+-+}.

(f) While the degree of m vertices in the graph G is $\deg_{G^{+-+}}(v) = 2n$, the degree of n vertices is $\deg_{G^{+-+}}(v) = 2m$. Since an edge in the graph G is connected to $(m-1)+(n-1)$ edges, each vertex in the graph $L(G)$ is adjacent to $(m-1)+(n-1)$ vertices. Therefore, in the graph $\overline{L(G)}$, each vertex is adjacent to $mn-1-(m+n-2)$ vertices. Therefore, for $\forall uv \in V(\overline{L(G)})$ $\deg_{G^{+-+}}(uv) = mn-m-n+3$. The graph's vertices have an eccentricity value of 2.

With the results we found, we get, for $m, n \geq 2$

$$\begin{aligned}\varepsilon^c\left(G^{++}\right) &= \sum \deg(v)e(v) \\ &= m \cdot 2n \cdot 2 + n \cdot 2m \cdot 2 + m \cdot n(mn-m-n+3) \cdot 2 \\ &= 2mn(mn-m-n+7).\end{aligned}$$

Furthermore, since the graph's vertices have an eccentricity value of 2, according to Theorem 1, $\varepsilon^c(G^{++}) = 4q$ where q is the number of edges of the graph G^{+-+}.

The theorem is thus proved. □

Theorem 3 *When $xyz = -++$, let the transformation graph of the graph G be G^{-++} and q is the number of edges of the graph G^{-++}.*

(a) If $G \cong P_n$ $(n \geq 6)$, then $\varepsilon^c(G^{-++}) = 2n^2 + 10n - 18$;

(b) If $G \cong C_n$ $(n \geq 6)$, then $\varepsilon^c(G^{-++}) = 2n^2 + 10n$;

(c) If $G \cong K_n$ $(n \geq 3)$, then $\varepsilon^c(G^{-++}) = 2n^2(n-1) = 4q$;

(d) If $G \cong W_{1,n}$ $(n \geq 3)$, then $\varepsilon^c(G^{-++}) = 4n^2 + 26n$;

(e) If $G \cong K_{m,n}$ $(m, n \geq 2)$, then $\varepsilon^c(G^{-++}) = 2(m+n)(mn+m+n-1) = 4q$.

Proof. (a) Let x and y be the pendant vertices of the graph G. It is easily seen that for $\forall v \in V(\bar{G})$ $\deg_{G^{++}}(v) = n-1$. Furthermore, $\deg_{G^{-++}}(xN_G(x)) = \deg_{G^{-++}}(yN_G(y)) = 3$ and for $uv \in V(\overline{L(G)}) - \{xN_G(x), yN_G(y)\}$ $\deg_{G^{-++}}(uv) = 4$. The eccentricity value of the vertices of the graph is calculated according to the vertices as follows.

- $e(u)$ value of $\forall u \in V(\bar{G})$: Let $A = N_{\bar{G}}(u), B = V(\bar{G}) - N_{\bar{G}}[u], C = N_{L(G)}(u)$ and $D = V(L(G)) - C$. The shortest distance between the vertex u and the vertices in $A \cup C$ and in $B \cup D$ is 1 and 2, respectively. Thus, we get $e(u) = 2$.

- $e(uv)$ value of $\forall uv \in V(L(G))$: Let $A = N_{\bar{G}}(uv), B = V(\bar{G}) - N_{\bar{G}}(uv)$ and $C = N_{L(G)}(uv)$. The shortest distance between the vertex uv and the vertices in $A \cup C$, in B and in $V(L(G)) - \left\{N_{L(G)}[uv] \cup N^2_{L(G)}(uv)\right\}$ is 1, 2 and 3, respectively. Thus, we get $e(uv) = 3$.

  With the results we found, we get, for $n \geq 6$, we have

$$\begin{aligned}\varepsilon^c\left(G^{-++}\right) &= \sum \deg(v)e(v) = n\cdot(n-1)\cdot 2 + 2\cdot 3\cdot 3 + (n-1-2)\cdot 4\cdot 3\\ &= 2n^2 + 10n - 18.\end{aligned}$$

(b) It is easily seen that for $\forall v \in V(\bar{G})$ $\deg_{G^{-++}}(v) = n-1$ and for $\forall uv \in L(P_n)$ $\deg_{G^{-++}}(uv) = 4$. As in Theorem 2 (a), we get eccentricity value for $\forall u \in V(\bar{G})$ and $\forall uv \in V(L(G))$ is $e(u) = 2$ and $e(uv) = 3$, respectively.

With the results we found, we get, for $n \geq 6$, we have

$\varepsilon^c(G^{-++}) = \sum \deg(v)e(v) = n(n-1)2 + n\cdot 4.3 = 2n^2 + 10n.$

(c) We can easily observe that for $\forall v \in V(\bar{G})$ $\deg_{G^{-++}}(v) = n-1$ and for all $uv \in L(G)$ $\deg_{G^{-++}}(uv) = 2n-2$. It is also seen that he eccentricity value of $\forall u \in V(G^{-++})$is $e(u) = 2$. With the results we found, we get, for $n \geq 5$,

$\varepsilon^c(G^{-++}) = \sum \deg(v)e(v) = n(n-1)2 + \frac{n(n-1)}{2}2(n-1)2 = 2n^2(n-1).$

Furthermore, since the graph's vertices have an eccentricity value of 2, according to Theorem1, $\varepsilon^c(G^{-++}) = 4q$ where q is the number of edges of the graph G^{-++}.

(d) Let c be the central vertex of the graph G. It is easy to see that $\deg_{G^{-++}}(c) = n$ and for $\forall v \in V(\bar{G}) - \{c\}\deg_{G^{-++}}(v) = n$. For the vertices corresponding to the edges connecting the central vertex and the vertices on the cycle graph, $\deg_{G^{-++}}(cN_G(c)) = n+3$ and for $\forall uv \in V(L(G)) - \{cN_G(c)\}\deg_{G^{-++}}(uv) = 6$. The graph's vertices have an eccentricity value of 2.

With the results we found, we get, for $n \geq 3$,

$$\varepsilon^c(G^{-++}) = \sum \deg(v)e(v) = (n+1)\cdot n\cdot 2 + n(n+3)2 + n\cdot 6\cdot 3 = 4n^2 + 26n.$$

(e) In the graph $\bar{G}$, the degree of each vertex v is $\deg_{G^{-++}}(v) = m + n - 1$. Since an edge in the graph G is connected to $(m-1)+(n-1)$ edges, each vertex in the graph $L(G)$ is adjacent to $(m-1)+(n-1)$ vertices. Therefore, for $\forall uv \in V(L(G))$ $\deg_{G^{-++}}(uv) = (m-1)+(n-1)+2 = m+n$. The graph's vertices have an eccentricity value of 2.

With the results we found, we get, for $m, n \geq 2$,

$\varepsilon^c(G^{-++}) = \sum \deg(v)e(v) = 2(m+n)(mn+m+n-1)$.

Furthermore, since the graph's vertices have an eccentricity value of 2, according to Theorem 1, $\varepsilon^c(G^{-++}) = 4q$ where q is the number of edges of the graph G^{-++}.

The theorem is thus proved. □

Theorem 4 *When $xyz = --+$, let the transformation graph of the graph G be G^{--+}, and q is the number of edges of the graph G^{--+}.*

(a) If $G \cong P_n$ $(n \geq 3)$, then $\varepsilon^c(G^{--+}) = 4\left(n^2-2n+2\right) = 4q$;

(b) If $G \cong C_n$ $(n \geq 4)$, then $\varepsilon^c(G^{--+}) = 4\left(n^2-n\right) = 4q$;

(c) If $G \cong K_n$ $(n \geq 4)$, then $\varepsilon^c(G^{--+}) = n(n-1)(n-7)(n+2)/2 = 4q$;

(d) If $G \cong K_{1,n}$ $(n \geq 3)$, then $\varepsilon^c(G^{--+}) = 2n^2+6n = 4q$;

(e) If $G \cong W_{1,n}$ $(n \geq 4)$, then $\varepsilon^c(G^{--+}) = 8n^2-4n = 4q$.

Proof. (a) Let x and y be the pendant vertices of the graph G. It is easily seen that for $\forall v \in V(\bar{G})$ $\deg_{G^{--+}}(v) = n-1$. Furthermore, $\deg_{G^{--+}}(xN_G(x)) = \deg_{G^{--+}}(yN_G(y)) = (n-2-1)+2 = n-1$ and for $\forall uv \in V(\overline{L(G)}) - \{xN_G(x), yN_G(y)\}\deg_G(uv) = n-2$. The graph's vertices have an eccentricity value of 2.

With the results we found, we get, for $n \geq 3$, we have

$$\begin{aligned}\varepsilon^c\left(G^{--+}\right) &= \sum \deg(v)e(v) = n\cdot(n-1)\cdot 2 + 2\cdot(n-1)\cdot 2 + (n-3)(n-2)2 \\ &= 4\left(n^2-2n+2\right).\end{aligned}$$

(b) We can easily observe that for $\forall v \in V(G^{--+})\deg_{G^{--+}}(v) = n-1$. The graph's vertices have an eccentricity value of 2. With the results we found, we get, for $n > 3$,

$\varepsilon^c(G^{--+}) = \sum \deg(v)e(v) = 2n(n-1)2 = 4n(n-1)$.

(c) We get for $\forall v \in V(\bar{G})$ $\deg_{G^{--+}}(v) = n-1$ and $\forall uv \in V(\overline{L(G)})$ $\deg_{G^{--+}}(uv) = \left(n^2-5n+10\right)/2$. The graph's vertices have an eccentricity value of 2. With the results we found, we get, for $n > 3$,

$$\begin{aligned}\varepsilon^c\left(G^{--+}\right) &= \sum \deg(v)e(v) = n(n-1)2 + n(n-1)/2.\left(n^2-5n+10\right)/2.2\\ &= n(n-1)(n-7)(n+2)/2.\end{aligned}$$

(d) Let c be the central vertex of the graph G. It is easy to see that $\deg_{G^{--+}}(c) = n$ and for $\forall v \in V(\bar{G}) - \{c\}\deg_{G^{--+}}(v) = n$. Since $\overline{L(K_{1,n})}$ contains n isolated peaks, for $\forall uv \in V(\overline{L(G)})\deg_{G^{--+}}(uv) = 2$. The graph's vertices have an eccentricity value of 2. With the results we found, we get, for $n \geq 3$,

$$\varepsilon^c\left(G^{--+}\right) = \sum \deg(v)e(v) = (n+1)\cdot n\cdot 2 + n\cdot 2\cdot 2 = 2n^2+6n.$$

(e) Let c be the central vertex of the graph G. It is easy to see that $\forall v \in V(\bar{G})$ $\deg_{G^{--+}}(v) = n$. For the vertices corresponding to the edges connecting the central vertex and the vertices on the cycle graph, $\deg_{G^{--+}}(cN_G(c)) = n$ and for $\forall uv \in V(\overline{L(G)}) - \{cN_G(c)\}$ $\deg_{G^{--+}}(uv) = 2n-3$. The graph's vertices have an eccentricity value of 2.

With the results we found, we get, for $n \geq 4$,

$$\varepsilon^c\left(G^{--+}\right) = \sum \deg(v)e(v) = (n+1)\cdot n\cdot 2 + n\cdot n\cdot 2 + n\cdot(2n-3)2 = 2n(4n-2).$$

Because of the form of the graph G^{--+} $(G \cong P_n, C_n, K_n, K_{1,n}, W_{1,n})$, it can be easily seen from above that the graph's vertices have an eccentricity value of 2. According to Theorem 1, we get $\varepsilon^c\left(G^{--+}\right) = 4q$ where q is the number of edges of the graph G^{--+}.

The theorem is thus proved. □

Theorem 5 *When* $xyz = +++$*, let the transformation graph of the graph* G *be* G^{+++} *and* q *is the number of edges of the graph* G^{+++}*.*

(a) If $G \cong C_n$ $(n \geq 3)$*, then* $\varepsilon^c\left(G^{+++}\right) = 8n\left\lceil \frac{n}{2} \right\rceil$*;*

(b) If $G \cong P_n$ $(n \geq 5)$*, then* $\varepsilon^c\left(G^{+++}\right) = 6(n-1)^2$*;*

(c) If $G \cong K_n$ $(n \geq 3)$*, then* $\varepsilon^c\left(G^{+++}\right) = 2\left(n^2+n\right)(n-1) = 4q$*;*

(d) If $G \cong K_{1,n}$ $(n \geq 3)$*, then* $\varepsilon^c\left(G^{+++}\right) = 2n^2+8n$*;*

(e) If $G \cong W_{1,n}$ $(n \geq 5)$*, then* $\varepsilon^c\left(G^{+++}\right) = 2n^2+46n$*.*

Proof. (a) The graph G^{+++} is a 4-regular graph. We get that the graph's vertices have an eccentricity value of $\lceil \frac{n}{2} \rceil$. With the results we found, we get, for $n \geq 3$, $\varepsilon^c(G^{++}) = \sum \deg(v)e(v) = 8n\lceil \frac{n}{2} \rceil$.

(b) Let x and y be the pendant vertices of the graph G. It is easily seen that $\deg_{G^{+++}}(x) = \deg_{G^{+++}}(y) = 2$ and for $\forall v \in V(G) - \{x, y\}$ $\deg_{G^{+++}}(v) = 4$. Furthermore, $\deg_{G^{+++}}(xN_G(x)) = \deg_{G^{+++}}(yN_G(y)) = 3$ and for $\forall uv \in V(L(G)) - \{xN_G(x), yN_G(y))$ $\deg_{G^{+++}}(uv) = 4$. There occur two cases depending on n for the graph's vertices' eccentricity values.

To make the proof clearer, let the n vertices of G be $v_1, v_2, v_3 \ldots v_{n-1,}, v_n$ and the $n-1$ vertices of $L(G)$ be $v_1v_2, v_2v_3, v_3v_4, \ldots, v_{n-1}v_n$.

Case 1. For n even. By the definition of the eccentricity value, for every vertex of G, it can easily be observed that,

$$e\left(v_{\frac{n}{2}}\right) = e\left(v_{\frac{n}{2}+1}\right), e\left(v_{\frac{n}{2}-1}\right) = e\left(v_{\frac{n}{2}+2}\right), \ldots, e(v_2) = e(v_{n-1}), e(v_1) = e(v_n).$$

Thus, we have $e(v_i) = e\left(v_{n-(i-1)}\right) = n - i$, where $i \in \{1, 2, \ldots, n/2\}$.
For every vertex of $L(G)$, we receive the following equalities.

$$e\left(v_{\frac{n}{2}-1}v_{\frac{n}{2}}\right) = e\left(v_{\frac{n+2}{2}}v_{\frac{n+4}{2}}\right), e\left(v_{\frac{n-4}{2}}v_{\frac{n-2}{2}}\right) = e\left(v_{\frac{n+4}{2}}v_{\frac{n+6}{2}}\right), \ldots, e(v_2v_3) = e(v_{n-2}v_{n-1}), e(v_1v_2) = e(v_{n-1}v_n).$$ These value are $e(v_jv_{j+1}) = e\left(v_{n-j}v_{n-(j-1)}\right) = n - j$, where $j \in \{1, 2, \ldots, (n/2) - 1\}$ and $e\left(v_{\frac{n}{2}}v_{\frac{n}{2}+1}\right) = n - \frac{n}{2}$.

With the results we found, we get, for $n \geq 6$,

$$\begin{aligned}\varepsilon^c\left(G^{+++}\right. &= 2\left(2(n-1) + \sum_{i=n/2}^{n-2} 4i\right) + 2\left(3(n-1) + \sum_{j=(n+2)/2}^{n-2} 4j\right) + 4\frac{n}{2} \\ &= 6n^2 - 12n + 6 = 6(n-1)^2.\end{aligned}$$

Case 2. For n odd. The eccentricity values for every vertex of G are

$$e\left(v_{\frac{n-1}{2}}\right) = e\left(v_{\frac{n+1}{2}+1}\right), e\left(v_{\frac{n-3}{2}}\right) = e\left(v_{\frac{n+1}{2}+2}\right), \ldots, e(v_2) = e(v_{n-1}),$$

$e(v_1) = e(v_n)$.

It is easy to see that $e(v_i) = e\left(v_{n-(i-1)}\right) = n - i$ where $i \in \left\{1, 2, \ldots, \frac{n-1}{2}\right\}$ and $e\left(v_{\frac{n+1}{2}}\right) = n - \frac{n+1}{2}$. Since the vertices in the $L(G)$ subgraph are even with degrees, the eccentricity values of the vertices are as in Case 1.

With the results we found, we get, for $n \geq 5$,

$$\varepsilon^c\left(G^{+++}\right) = 2\left(2(n-1)+\sum_{i=(n+1)/2}^{n-2} 4i\right)+4\frac{n-1}{2}+2\left(3(n-1)+\sum_{i=(n+1)/2}^{n-2} 4i\right)$$
$$= 6n^2-12n+6 = 6(n-1)^2.$$

(c) We get for $\forall v \in V(G)\ \deg_{G^{+++}}(v)=2(n-1)$ and $\forall uv \in V(L(G))\ \deg_{G^{+++}}(uv) = 2(n-1)$. The graph's vertices have an eccentricity value of 2. With the results we found, we get, for $n \geq 3$,

$$\varepsilon^c\left(G^{+++}\right) = \sum \deg(v)e(v) = 2\left(n^2+n\right)(n-1).$$

Furthermore, since the graph's vertices have an eccentricity value of 2, according to Theorem 1, $\varepsilon^c\left(G^{+++}\right) = 4q$ where q is the number of edges of the graph G^{+++}.

(d) Let c be the central vertex of the graph G. It is easy to see that $\deg_{G^{+++}}(c) = 2n$ and for $\forall v \in V(G)-\{c\}\quad \deg_{G^{+++}}(v) = 2$. Since the structure of the $L(G)$ subgraph is a complete graph with n vertices, for $\forall uv \in V(L(G))\ \deg_{G^{+++}}(uv) = n+1$. Also, it is easily seen that $e(c) = 1$ and for $\forall v \in V\left(G^{+++}\right)-\{c\}\quad e(v) = 2$.

With the results we found, we get, for $n \geq 3$,

$$\varepsilon^c\left(G^{+++}\right) = \sum \deg(v)e(v) = 2n+n\cdot 2\cdot 2+n\cdot(n+1)2 = 2n^2+8n.$$

(e) The vertex set of G^{+++} can be partitioned into four subsets as

$V_1\left(G^{+++}\right)$; central vertex c of G. $\deg_{G^{+++}}(c) = 2n$.

$V_2\left(G^{+++}\right) = V(G)-\{c\}$. For $\forall u \in V_2\left(G^{+++}\right), \deg_{G^{+++}}(u) = 6$.

$V_3\left(G^{+++}\right)$: the $\{cN_G(c)\}$ vertices in $L(G)$. For $\forall xy \in V_3\left(G^{+++}\right)\deg_{G^{+++}}(xy) = n+3$.

Furthermore, the vertices of $V_3\left(G^{+++}\right)$ are a complete graph in themselves.
$V_4\left(G^{+++}\right)$; the vertices in $L(G)$ formed by the edges of the graph C_n. For $\forall xy \in V_4\left(G^{+++}\right)\deg_{G^{+++}}(xy) = 6$.

The eccentricity values of the vertices are as follows: If the central vertex c, then for $\forall u \in V_2(G^{+++})$ $d(c,u) = 1$, for $\forall xy \in V_3(G^{+++})$ $d(c,xy) = 1$ and for $\forall xy \in V_4(G^{++})$ $d(c,xy) = 2$. Thus, we get $e(c) = 2$.

If $v \in V_2(G^{+++})$, then $d(c,v) = 1$, for $\forall u \in V_2(G^{+++}) - N_G[v]$ $d(v,u) = 2$, for $\forall xy \in V_3(G^{++})$ $d(v,xy) \leq 2$ and for $\forall xy \in V_4(G^{+++})$ $d(v,xy) \leq 3$. Hence, we have $e(v) = 3$. Similarly, we have $e(xy) = 2$ for $\forall xy \in V_3(G^{+++})$ and $e(xy) = 3$ for $\forall xy \in V_4(G^{+++})$. With the results we found, we get, for $n \geq 5$, $\varepsilon^c(G^{+++}) = \sum \deg(v)e(v) = 4n + 18n + n(n+3)2 + 18n = 2n^2 + 46n$. □

References

[1] A. Aytaç and T. Turacı, Vulnerability Measures of Transformation Graph, *Inter. Journal of Found. of Comp. Sci.*, **26**, 6 (2015), 667–675. ⇒113

[2] A. T. Balaban, Highly discriminating distance-based topological index, *Chem. Phys. Lett.*, **89**, (1982), 399–404. ⇒112

[3] Z.N. Berberler and M.E. Berberler, Edge eccentric connectivity index of nanothorns, *Bulgarian Chemical Communications*, **48**, 1 (2016), 165–170 ⇒112

[4] Z.N. Berberler and M.E. Berberler, The Edge Geometric-Arithmetic Index of an Infinite Class of Dendrimers, *Journal of Computational and Theoretical Nanoscience*, **10**, 9 (2013), 2207–2208. ⇒112

[5] Z.N. Berberler and M.E. Berberler, The Modified Eccentric Connectivity Index of Nanocomposites, *Journal of Computational and Theoretical Nanoscience*, **12**, 9 (2015), 2208–2212. ⇒112

[6] W. Gao and M. R. Farahani, Degree-based indices computation for special chemical molecular structures using edge dividing method, *Appl. Math. Nonlinear Sci.* **1**, (2016), 94–117. ⇒111

[7] W. Gao, W. F. Wang and Farahani M. R., Topological indices study of molecular structure in anticancer drugs, *Journal of Chemistry*, (2016). ⇒111

[8] S. Gupta, M. Singh and A. K. Madan, Application of Graph Theory: Relationship of Eccentric Connectivity Index and Wiener's Index with Anti-inflammatory Activity, *J. Math. Anal. Appl.*, **266**, 2 (2002), 259–268. ⇒111

[9] W. Harry, Structural Determination of Paraffin Boiling Points , *Journal of the American Chemical Society*, **69**, 1 (1947), 17–20. ⇒112

[10] H. Hosoya, Topological index. A newly proposed quantity characterizing the topological nature of structural isomers of saturated hydrocarbons, *Bulletin of the Chemical Society of Japan*, **44**, 9 (1971), 2332–2339. ⇒112

[11] O. Ivanciuc, T. Ivanciuc and A.T. Balaban, Vertex-and edge-weighted molecular graphs and derived structural descriptors, *Topological Indices and Related Descriptors in QSAR and QSPR*, CRC Press, 1999. ⇒111

[12] Z.N. Odabaş, The Edge Eccentric Connectivity Index of Dendrimers, *Journal of Computational and Theoretical Nanoscience*, **10**, 4 (2013), 783–784. ⇒112

[13] S. Sardana and A.K. Madan, Predicting anti-HIV activity of TIBO derivatives: a computational approach using a novel topological descriptor, *J. Mol. Model.*, **8**, (2002), 258–265. ⇒111

[14] V. Sharma, R. Goswami, A.K. Madan, Eccentric Connectivity Index: A Novel Highly Discriminating Topological Descriptor for Structure-Property and Structure-Activity Studies, *J. Chem. Inf. Comput. Sci.*, **37**, (1997), 273–282. ⇒112

[15] R. Todeschini and V. Consonni, *Handbook of Molecular Descriptors*, Wiley-VCH, Weinheim, 2008. ⇒111

[16] D.B. West, *Introduction to Graph Theory*, Prentice Hall, NJ, 2001. ⇒112

[17] B. Wu and J. Meng, Basic properties of total transformation graphs, *J. Math. Study*, **34**, 2 (2001), 109–116. ⇒113

[18] B. Wu, L. Zhang and Z. Zhang, The transformation graph G^{xyz} when $xyz = -++$, *Discrete Mathematics*, **296**, (2005), 263–270. ⇒113

[19] L. Xu and B. Wu, Transformation graph, *Discrete Mathematics*, **308**, (2008), 5144–5148. ⇒113

[20] X. Xu and Y. Guo, The Edge Version of Eccentric Connectivity Index, *International Mathematical Forum*, **7(6)**, (2012), 273–280 . ⇒112

[21] L. Xueliang and S. Yongtang, A survey on the Randic index, *MATCH Communications in Mathematical and in Computer Chemistry*, **59**, 1 (2008), 127–156. ⇒112

[22] B. Zhou and Z. Du, On eccentric connectivity index, *MATCH Commun. in Math.and in Computer Chem.*, **63**, (2010), 181–198. ⇒113

Received: January 18, 2023 • Revised: July 3, 2023

 Acta Univ. Sapientiae Informatica **15,** 1 (2023) 124–145

DOI: 10.2478/ausi-2023-0010

On agglomeration-based rupture degree in networks and a heuristic algorithm

Muammer AĞTAŞ
Pamukkale University
Faculty of Engineering
Department of Computer Engineering
20160 Denizli, Turkey
email: magtas21@posta.pau.edu.tr

Tufan TURACI
Pamukkale University
Faculty of Engineering
Department of Computer Engineering
20160 Denizli, Turkey
email: tturaci@pau.edu.tr

Abstract. The rupture degree is one the most important vulnerability parameter in networks which are modelled by graphs. Let $G(V(G),E(G))$ be a simple undirected graph. The rupture degree is defined by $r(G) = \max\{w(G–S)–|S|–m(G–S):S\subset V(G)$ and $w(G–S)>1\}$ where $m(G–S)$ is the order of a largest connected component in $G–S$ and $w(G–S)$ is the number of components of $G–S$, respectively. In this paper, we consider the vertex contraction method based on the network agglomeration operation for each vertex of G. Then, we have presented two graph vulnerability parameters called by *agglomeration rupture degree* and *average lower agglomeration rupture degree*. Furthermore, the exact values of them for some graph families are given. Finally, we proposed a polynomial time heuristic algorithm to obtain the values of *agglomeration rupture degree* and *average lower agglomeration rupture degree*.

1 Introduction

Networks can be modeled with graphs. The servers or hubs are illustrated by vertices and edges are connecting medium between them in any graph G. The

Key words and phrases: graphs, network design and communication, complex networks, connectivity, vulnerability, rupture degree, agglomeration.

vulnerability of a network is of main significance to network planners according to the nodes and links [7, 12]. Recently, networks vulnerability has been studied in widespread multidisciplinary area such as informatics, mathematics, computer science, chemistry and many other applied science and engineering science. The vulnerability value of networks is defined as the durability of the network after the breakdown of some vertices or edges until a communication disruption [12, 22].

In this paper, we consider only simple graphs. Now, some notations will be given. Let $G(V(G),E(G))$ be a simple connected graph whose vertex and edge sets are denoted by $V(G)$ and $E(G)$, where $V(G)=\{v_1,v_2,\ldots,v_n\}$, $E(G)=\{e_1,e_2,\ldots,e_m\}$, $|V(G)|=n$ and $|E(G)|=m$. Let $u\in V(G)$.The set $N(u)=\{v\in V(G)|(u,v)\in E(G)\}$ is called the *open neighborhood* of u. Furthermore, the number of $|N(u)|$ is called the degree of vertex u and is denoted by $d_G(u)$. The *maximum degree* of G is denoted by $\Delta(G)$ is defined by $\max\{d_G(v) \mid v\in V(G)\}$. Similarly, the *minimum degree* of G is denoted by $\delta(G)$ is defined by $\min\{d_G(v) \mid v\in V(G)\}$ [18, 33]. The set $N[u]=\{u\}\cup N(u)$ is called the *closed neighborhood* of the vertex u. The $d(u,v)$ represents the distance between two vertices as u and v. Furthermore, the distance is defined as the length of a shortest path between them [18, 33].

The connectivity value of any graph G is the best-known vulnerability measures in the literature. It is defined that to obtain disconnected graph after the minimum number of vertices are deleted from the given graph, also is denoted by $k(G)$ for any graph G [16]. The connectivity of any graph G is computed with polynomial time. There are many vulnerability measures for the networks. For example, integrity [9], toughness [12], tenacity [13], global distribution number [14] are considered and studied in many areas. Furthermore, there are many average vulnerability parameters are proposed to obtain the vulnerability values of the networks. For example, the average lower domination number [17], the average lower independence number [6], the average lower bondage number [32], the average lower reinforcement number [31], the average lower residual domination number [29], the average lower link residual domination numbers [30] etc. are considered and studied in many areas. The values of these parameters are not computed in polynomial time. Because they are classes of NP-Hard or NP-Complete.

The rupture degree is other best known vulnerability parameter. It is defined by Li et al. in [23] and the definition of it as the following:

$$r(G) = \max\{w(G–S)–|S|–m(G–S):S\subset V(G) \text{ and } w(G–S)>1\},$$

where $m(G–S)$ and $w(G–S)$ denote the is the largest connected component in $G–S$ and the number of components of the graph $G–S$, respectively.

Let C_6 be a cycle graph. It is showed by in Figure 1. The alternative rupture-sets of C_6 are showed with the set of darkened vertices. Clearly, $|S|$=3, $w(C_6 - S) = 3$ and $m(C_6 - S) = 1$. As a result,$r(C_6)$=1 is obtained.

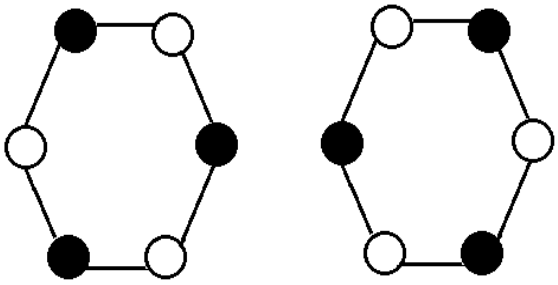


Figure 1: The rupture-set of the graph C_6

In [26], the authors showed that calculating the rupture degree problem is an NP-complete problem. However, it is possible to determine the rupture degree of large classes of graphs. For more results on rupture degree, we refer the readers to see [1, 2, 3, 4, 5, 8, 20, 21, 25, 27]. Furthermore, Li gave an algorithm whose complexity is $O(n^2)$ for isolating rupture degree in Trees of order n [24]. Another interesting study about the rupture degree is the references [15] by Durgut et al. In [15], a heuristic algorithm is given to find the rupture degree for any graph G. A similar study is in [11], where Berberler et al. gave a polynomial time heuristic algorithm for computing the integrity of any given graph G.

When investigating the vulnerability of complex networks, the finding node importance is used for each node recently. There are some different methods for determining importance of each node. In this paper, we use node contraction method based on network agglomeration. Then, new two vulnerability parameter definitions have been made by combining node contraction method based on network agglomeration method and the rupture degree. By using methods based on agglomeration, more efficient results can be obtained in terms of vulnerability. Let $v_i \in V(G)$. The node contraction is defined as follows: the node v_i and other $d_G(v_i)$ nodes connected with v_i into a new node $v_i^{'}$, which takes place of the primary $d_G(v_i)$+1 nodes, and links connected with

$d_G(v_i)$–1 nodes originally turn to the new node $v_i^{'}$ now. For example, if the center node is contracted in a star-network, the network is agglomerated to one node. Another example can be seen in Figure 2.

The agglomeration operation has been used in different network vulnerability measures, some of these can be seen in [10, 19, 28]. In this paper, we incorporate the concept of the rupture degree and agglomeration operation, as well as the idea of the average vulnerability parameters, to introduce new graph parameters called the agglomeration rupture degree (ARD), denoted by $r^{agg}(G)$, and the average lower agglomeration rupture degree (ALARD), denoted by $r_{av}^{agg}(G)$, for any given graph G. Furthermore, we consider the ARD and ALARD to be two metrics for network vulnerability.

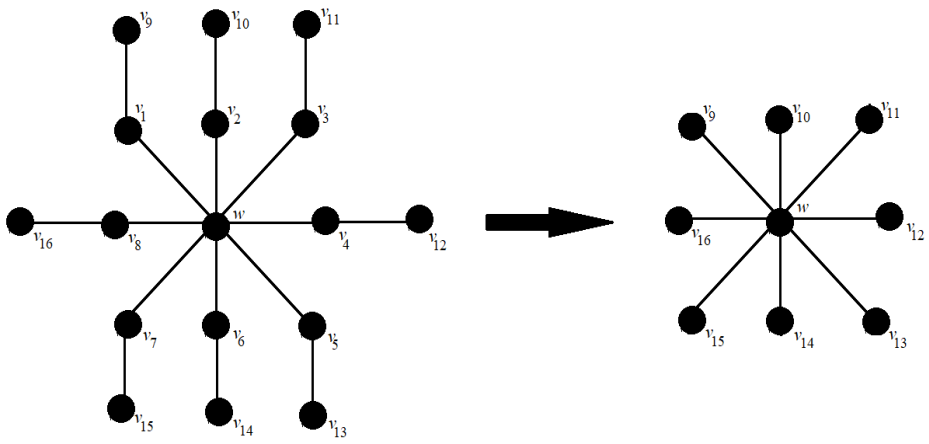


Figure 2: The agglomeration operation on the vertex w.

In this paper, there are 6-Sections. The ARD and ALARD are defined in Section 2. In Section 3, the difference of the ARD and ALARD is shown with different examples. The values of ARD and ALARD are obtained some well-known graph families in Section 4. In Section 5, we give a polynomial time heuristic algorithm to compute the values of ARD and ALARD, then the computational test results are presented. Finally, we give our conclusions in Section 6.

2 The definitions of the ARD and ALARD

The definitions of ARD and ALARD are given in this section. For a vertex v_k of a graph G, the *lower agglomeration rupture degree*, denoted by $r^{agg}_{v_k}(G)$, is the minimum cardinality of the *rupture set* in G derived from the graph G after the agglomeration operation for the vertex v_k. The *agglomeration rupture degree* of a graph G is defined as:

$$r^{agg}(G) = max_{v_k \in V(G)}\{r^{agg}_{v_k}(G)\}.$$

Furthermore, the *average lower rupture degree* of G is defined by

$$r^{agg}_{av}(G) = \frac{1}{|V(G)|} \sum_{v_k \in V(G)} r^{agg}_{v_k}(G).$$

Example 1.1. Let the graph G, which are showed in Figure3, be a graph with 6-vertices and 6-edges. Clearly, the connectivity number and the rupture degree of G is one. The rupture set of G is $\{v_1,v_4\}$ and $r(G)$=1.

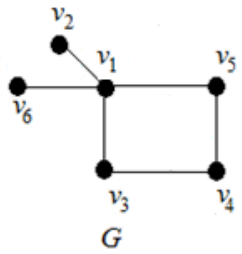


Figure 3: The graph G whose number of vertices and edges is 6.

Vertices	$r^{agg}_{v_k}(G)$
v_1	-1
v_2	0
v_3	1
v_4	1
v_5	1
v_6	0

Table 1: The lower agglomeration rupture degree of every vertex $v_k \in V(G)$

The lower agglomeration rupture degree of every vertex $v_k \in V(G)$ is presented in Table 1.

Clearly, we have $r_{v_1}^{agg}(G)$=-1, $r_{v_2}^{agg}(G)$=0, $r_{v_3}^{agg}(G)$=1, $r_{v_4}^{agg}(G)$=1, $r_{v_5}^{agg}(G)$=1, and $r_{v_6}^{agg}(G)$=0. Thus, $r^{agg}(G)$=1 and $r_{av}^{agg}(G)$=(-1+0+1+1+1+0)/6=2/6=0.33 are obtained.

3 Vulnerability examples of the ARD and ALARD

The ARD and ALARD can be more efficient than the connectivity and the rupture degree in measuring the vulnerability of some graphs. In this section, this situation is showed with different two examples.

In the first example, we consider the graphs G_1 and G_2 that are presented in Figure 4. Then, we want to show the values of ARD and ALARD can be used to distinguish between two given graphs. Clearly, the values of connectivity and domination number, and also the numbers of vertices and edges of the graphs G_1 and G_2 are equal. That is, $k(G_1)$=$k(G_2)$=1, $r(G_1)$=$r(G_2)$=1, $|V(G_1)|$=$|V(G_2)|$=8 and $|E(G_1)|$=$|E(G_2)|$=8.

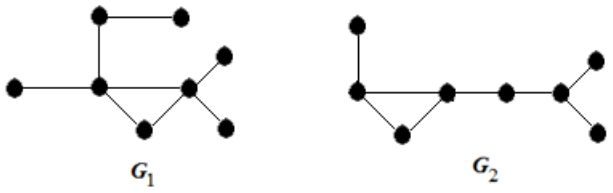


Figure 4: The graphs G_1 and G_2 with 8-vertices and 8-edges

The ARD of the graphs G_1 and G_2 are $r^{agg}(G_1)$=2 and $r^{agg}(G_2)$=1, while the ALARDs of these two graphs G_1 and G_2 are $r_{av}^{agg}(\mathrm{G_1}) = \frac{1}{2}$ and $r_{av}^{agg}(\mathrm{G_2}) = \frac{1}{4}$, respectively.

In the second example, we consider the graphs G_3 and G_4 that are presented in Figure 5. Then, we want to show the value of ALARD can be used to distinguish between two given graphs. Clearly, the values of connectivity, the rupture degree and the agglomeration rupture degree of the graphs G_3 and G_4 are equal, with $k(G_3)$=$k(G_4)$=1, $r(G_3)$=$r(G_4)$=1 and $r^{agg}(G_3)$= $r^{agg}(G_4)$=1. Additionally, the numbers of vertices and edges of the graphs G_3 and G_4 are equal as like $|V(G_3)|$=$|V(G_4)|$=6 and $|E(G_3)|$=$|E(G_4)|$=6.

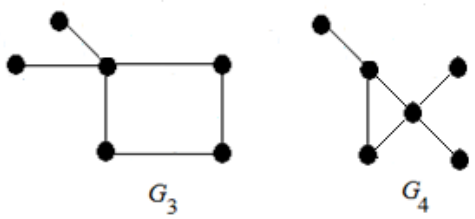


Figure 5: The graphs G_3 and G_4 with 6-vertices and 6-edges

The ALARD of the graphs G_3 and G_4 are $r_{av}^{agg}(\mathrm{G}_3) = \frac{1}{3}$ and $r_{av}^{agg}(\mathrm{G}_4) = 0$, respectively.

With these examples, we can say that these two new parameters ARD and ALARD may be more distinctive than other vulnerability parameters.

4 Computing the ARD and ALARD of well-known graphs

In this section, we compute the values of ARD and ALARD of well-known graphs such as the path graph P_n, the cycle graph C_n , the complete graph K_n, the star graph $K_{1,n-1}$, the wheel graph $W_{1,n}$ and complete bipartite graph $K_{n,m}$.

Theorem 1 *Let $G \cong P_n$ be a path graph of order n, where $n \geq 4$. Then,*

$$\textbf{(a)}\ r^{agg}(P_n) = 0 \quad \textbf{(b)}\ r_{av}^{agg}(P_n) = \begin{cases} -2/n, & \text{if } n \text{ is odd;} \\ (2-n)/n, & \text{if } n \text{ is even.} \end{cases}$$

Proof. We know that $r(P_n)$ = -1 if n is even; $r(P_n) = 0$ if n is odd (see [23]), and let $\{v_1, v_2, \ldots, v_{n-1}, v_n\}$ be vertices of P_n. In here, we say that the vertices v_1 and v_n are minor vertices, remaining vertices are called major vertices. Clearly, number of minor and major vertices are 2 and n-2, respectively. While we are calculating the ARD and ALARD of the path graph P_n, we have two cases depending on n.

Case 1. Let n be even. We distinguish two sub cases depending on the vertices of P_n.

Subcase 1.1. If a minor vertex is agglomerated, then a path P_{n-1} is obtained. Due to n is even, we have $r(P_{n-1}) = 0$. So, we obtain $r_{v_1}^{agg}(G)$=0 and $r_{v_n}^{agg}(G)$=0.

Subcase 1.2. If a major vertex is agglomerated, then a path $P_{n\text{-}2}$ is obtained. Due to n is even, we have $r(P_{n\text{-}2})$ = -1. So, we obtain $r^{agg}_{v_k}(G)$=-1, where $k \in \{2,3,\ldots,n\text{-}1\}$.

Finally, we get $r^{agg}(P_n)$=0 by the definition of ARD and the Subcases 1.1 and 1.2.

Furthermore, we get

$$\begin{aligned} r^{agg}_{av}(G) &= \frac{1}{|V(G)|}\left(\sum_{v_k \in V(G)} r^{agg}_{v_k}(G)\right) \\ &= \frac{1}{n}\left(r^{agg}_{v_1}(G) + r^{agg}_{v_n}(G) + \sum_{k=2}^{n-1} r^{agg}_{v_k}(G)\right) \\ &= \frac{1}{n}\Big(2(0) + (-1(n-2))\Big) \\ &= \frac{2-n}{n}. \end{aligned}$$

Case 2. Let n be odd. We distinguish two sub cases depending on the vertices of P_n.

Subcase 2.1. If a minor vertex is agglomerated, then a path $P_{n\text{-}1}$ is obtained. Due to n is odd, we have $r(P_{n\text{-}1}) = -1$. So, we obtain $r^{agg}_{v_1}(G)$=-1 and $r^{agg}_{v_n}(G)$=-1.

Subcase 2.2. If a major vertex is agglomerated, then a path $P_{n\text{-}2}$ is obtained. Due to n is odd, we have $r(P_{n\text{-}2}) = 0$. So, we obtain $r^{agg}_{v_k}(G)$=0, where $k \in \{2,3,\ldots,n\text{-}1\}$.

Finally, we get $r^{agg}(P_n)$=0 by the definition of ARD and the Subcases 2.1 and 2.2.

Furthermore, we get

$$\begin{aligned} r^{agg}_{av}(G) &= \frac{1}{|V(G)|}\left(\sum_{v_k \in V(G)} r^{agg}_{v_k}(G)\right) \\ &= \frac{1}{n}\left(r^{agg}_{v_1}(G) + r^{agg}_{v_n}(G) + \sum_{k=2}^{n-1} r^{agg}_{v_k}(G)\right) \\ &= \frac{1}{n}\Big(2(-1) + (0(n-2))\Big) \\ &= \frac{-2}{n}. \end{aligned}$$

By the Cases 1 and 2, the proof is completed. □

Theorem 2 *Let $G \cong C_n$ be a cycle graph of order n, where $n \geq 5$. Then,*

$$r^{agg}(C_n) = r^{agg}_{av}(C_n) = \begin{cases} -2, & \text{if } n \text{ is odd;} \\ -1, & \text{if } n \text{ is even.} \end{cases}$$

Proof. We know that $r(C_n)$ = -1 if n is even; $r(C_n)$ = -2 if n is odd (see [23]), and let $\{v_1,v_2,\ldots,v_{n-1},v_n\}$ be vertices of C_n. If a vertex is agglomerated in the graph C_n, then a cycle C_{n-2} is obtained. We have two cases depending on n.

***Case* 1.** Let n be even. Due to n is even, we have $r(C_{n-2})$ = -1. So, we obtain $r^{agg}_{v_k}(G)$=-1, where $k \in \{1,2,\ldots,n\}$.

***Case* 2.** Let n be odd. Due to n is odd, we have $r(C_{n-2})$ = -2. So, we obtain $r^{agg}_{v_k}(G)$=-2, where $k \in \{1,2,\ldots,n\}$.

Finally, we get

$$r^{agg}(C_n) = r^{agg}_{av}(C_n) = \begin{cases} -2, & \text{if } n \text{ is odd;} \\ -1, & \text{if } n \text{ is even.} \end{cases}$$

By the Cases 1 and 2, the proof is completed. □

Theorem 3 *Let $G \cong K_n$ be a complete graph of order n, where $n \geq 3$. Then,*

$$r^{agg}(K_n) = r^{agg}_{av}(K_n) = 0.$$

Proof. The rupture degree of K_n is defined as $r(K_n)$=1−n [23]. Let $\{v_1,v_2,\ldots,v_{n-1},v_n\}$ be vertices of K_n. If a vertex is agglomerated in the graph K_n, then the graph K_1 is obtained. Clearly, $r(K_1)$=0. So, we get $r^{agg}_{v_k}(G)$=0, where $k \in \{1,2,\ldots,n\}$. Thus, $r^{agg}(K_n)$=$r^{agg}_{av}(K_n)$=0 is obtained. □

Theorem 4 *Let $G \cong K_{1,n-1}$ be a star graph of order n, where $n \geq 4$. Then,*

(a) $r^{agg}(K_{1,n-1}) = n-4$ ***(b)*** $r^{agg}_{av}(K_{1,n-1}) = \dfrac{n^2-5n+4}{n}$.

Proof. The rupture degree of $K_{1,n\text{-}1}$ is defined as $r(K_{1,n\text{-}1})=n\text{-}3$ [23]. Let $\{v_c,v_1,v_2,\ldots,v_{n\text{-}2},v_{n\text{-}1}\}$ be vertices of $K_{1,n\text{-}1}$, where the vertex v_c is the center vertex of $K_{1,n\text{-}1}$. We distinguish two cases depending on the vertices of $K_{1,n\text{-}1}$.

***Case* 1.** If the center vertex v_c is agglomerated, then the complete graph K_1 is obtained. We know $r(K_1)=0$ [23]. So, we obtain $r^{agg}_{v_c}(G)=0$.

***Case* 2.** If a vertex v_k, where $k \in \{2,3,\ldots,n\text{-}1\}$, is agglomerated, then a star graph $K_{1,n\text{-}1}$ is obtained. Thus, we get $r^{agg}_{v_k}(G)=n-4$ for $k \in \{2,3,\ldots,n\text{-}1\}$.

Finally, we have $r^{agg}(K_{1,n\text{-}1})=0$ by the definition of ARD and the Cases 1 and 2.

Furthermore, we get

$$\begin{aligned} r^{agg}_{av}(G) &= \frac{1}{|V(G)|}\left(\sum_{v_k\in V(G)} r^{agg}_{v_k}(G)\right) \\ &= \frac{1}{n}\left(r^{agg}_{v_c}(G)+\sum_{k=1}^{n-1} r^{agg}_{v_k}(G)\right) \\ &= \frac{1}{n}\Big((n-1)(n-4)\Big) \\ &= \frac{n^2-5n+4}{n}. \end{aligned}$$

By the Cases 1 and 2, the proof is completed. □

Theorem 5 *Let $G \cong W_{1,n}$ be a wheel graph of order $n+1$, where $n \geq 5$. Then,*

(a) $r^{agg}(W_{1,n})=0$ ***(b)*** $r^{agg}_{av}(W_{1,n}) = \begin{cases} -2n/(n+1), & \text{if } n \text{ is odd;} \\ -n/(n+1), & \text{if } n \text{ is even.} \end{cases}$

Proof. The rupture degree of $W_{1,n}$ is defined as $r(W_{1,n})$ = -2 if n is even, and $r(W_{1,n})$ = -3 if n is odd (see [23]). Let $\{v_c,v_1,v_2,\ldots,v_{n\text{-}1},v_n\}$ be vertices of $W_{1,n}$, where the vertex v_c is the center vertex of $W_{1,n}$. We distinguish two cases depending on the vertices of $W_{1,n}$.

Case 1. If the center vertex v_c is agglomerated, then the complete graph K_1 is obtained. We know $r(K_1)=0$ [23]. So, we obtain $r^{agg}_{v_c}(G)=0$.

Case 2. If a vertex v_k, where $k \in \{1,2,\ldots,n\}$, is agglomerated, then a join graph K_1+P_{n-3} is obtained. We distinguish two sub cases depending on the number of n.

Subcase 2.1. If n is even, then n-3 will be odd. Due to is $n-3$ odd, then we get $\mathrm{r}(K_1+P_{n-3})=-1$ [23]. That is $r^{agg}_{v_k}(G)=-1$, where $k \in \{1,2,\ldots,n\}$.

Subcase 2.2. If n is odd, then $n-3$ will be even. Due to is $n-3$ even, then we get $\mathrm{r}(K_1+P_{n-3})=-2$ [23]. That is $r^{agg}_{v_k}(G)=-2$, where $k \in \{1,2,\ldots,n\}$.

Finally, we get $r^{agg}(W_{1,n})=0$ by the definition of ARD and the Cases 1 and 2. Thus, we get

$$r^{agg}_{av}(G) = \frac{1}{|V(G)|}\left(\sum_{v_k \in V(G)} r^{agg}_{v_k}(G)\right) = \frac{1}{n}\left(r^{agg}_{v_c}(G) + \sum_{k=1}^{n-1} r^{agg}_{v_k}(G)\right)$$

$$= \frac{1}{n+1}(n)(-1) = \frac{-n}{n+1}, \quad \text{if } n \text{ is even.}$$

If n is odd, then we have

$$r^{agg}_{av}(G) = \frac{1}{|V(G)|}\left(\sum_{v_k \in V(G)} r^{agg}_{v_k}(G)\right) = \frac{1}{n}\left(r^{agg}_{v_c}(G) + \sum_{k=1}^{n-1} r^{agg}_{v_k}(G)\right)$$

$$= \frac{1}{n+1}(n)(-2) = \frac{-2n}{n+1}.$$

By the Cases 1 and 2, the proof is completed. □

Theorem 6 *Let $G \cong K_{n,m}$ be a complete bipartite graph of order $n+m$, where $1 < n \leq m$. Then,*

$$\textbf{(a)}\ r^{agg}(K_{n,m}) = m-3 \quad \textbf{(b)}\ r^{agg}_{av}(K_{n,m}) = \frac{m^2+n^2-3m-3n}{n+m}.$$

Proof. The rupture degree of $K_{n,m}$ is defined as $r(K_{n,m})=1-m-n$ [23].
Let $\{v_1,v_2,\ldots,v_n,v'_1,v'_2,\ldots,v'_n\}$ be vertices of $K_{n,m}$. We distinguish two cases depending on the vertices of $K_{n,m}$.

Case 1. If a vertex v_k, where $k \in \{1,2,\ldots,n\}$, is agglomerated, then a star graph $K_{1,n-1}$ is obtained. We have $r(K_{1,n-1})=n-3$. Thus, we get $r^{agg}_{v_k}(G)=n-3$ for $k \in \{1,2,\ldots,n\}$.

Case 2. If a vertex v'_k, where $k \in \{1,2,\ldots,m\}$, is agglomerated, then a star graph $K_{1,m-1}$ is obtained. We have $r(K_{1,m-1})=m-3$. Thus, we get $r^{agg}_{v_k}(G)=m-3$ for $k \in \{1,2,\ldots,m\}$.

We have $r(K_{n,m})=max\{n-3,m-3\}$. Due to $n\leq m$, we obtain $r(K_{n,m})=m-3$ by Cases 1 and 2.

Furthermore, we get

$$\begin{aligned} r_{av}^{agg}(G) &= \frac{1}{|V(G)|}\left(\sum_{v_k\in V(G)} r_{v_k}^{agg}(G)\right) \\ &= \frac{1}{n+m}\left(\sum_{k=1}^{n} r_{v_k}^{agg}(G) + \sum_{k=1}^{m} r_{v_k'}^{agg}(G)\right) \\ &= \frac{1}{n+m}\Big((n)(n-3)+(m)(m-3)\Big) \\ &= \frac{m^2+n^2-3m-3n}{n+m}. \end{aligned}$$

By the Cases 1 and 2, the proof is completed. □

5 A heuristic algorithm for computing the ARD and ALARD

In this section, firstly we give the pseudocode of heuristic algorithm for ARD and ALARD in Appendix A. This algorithm runs polynomial time to find the ARD and ALARD of an arbitrary graph G. We give an example how the proposed algorithm works on the following graph P_4.

Let P_4 be a path graph and the node array(labelled of nodes) is [0, 1, 2, 3] into the operation function. This graph showed in the Figure 6.

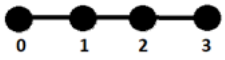


Figure 6: The graph P_4 whose vertices labelled by [0,1,2,3]

Let the vertex 1 and graph go to Agglomeration function in our Algorithm. The content of our neighbors array will be [[1], [0,2], [1,3], [2]]. For example, the content of the zeroth index is 1. So, the neighbor of node 0 will be 1. The content of aggCluster array is also [0,2]. Then we add the corresponding node and sort it from the largest number to the smallest number. The content would be [2, 1, 0]. Now we need to delete the row and column from the two-dimensional graph array. This process is also based on aggCluster. After

deletion, our components array is created, then we have components = [[0], [1], [2,3]] as like the following Figure 7.

Figure 7: The graph P_4 after the first deletion.

Now the neighbors are deleted from the array of components. In summary, this is the function of keeping vertices that are not adjacent to the *agglomeration vertex* in the array of components. Now, we have components = [[3]].

Then, labeledComponents = components has been made. Furthermore, tag_number is 1 in the loop. Since labeledComponents is single content, the loop returns one and labeledComponents [0][0] = label_number. In other words, label 1 is given to the neighbor of the merged nodes(0) and the newGraf becomes P_2 as like the following Figure 8.

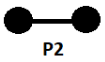


Figure 8: The graph P_2

The created newGraph is sent to the Rupture function which is proposed in [15] and then returns -1. The Agglomeration function also returns this Integer value. The Integer value from the operation function is added to the ruptures sequence. This event is made for all nodes and ruptures = [0, -1, -1, 0] is obtained. As a result, the maximum value will be ARD and its arithmetic average will be ALARD, that is $r^{agg}(P_4)$=0 and $r^{agg}_{av}(P_4) = \frac{-1}{2}$ are obtained.

5.1 Computational tests

In this section, the datasets of the references [11] and [15] have been used to perform our proposed algorithm. In the following tables, |V| is the number of vertices; ARD is the Heuristic result of Agglomeration Rupture Degree; ARDopt displays the Brute Force result of Agglomeration Rupture Degree; ALARD is the Heuristic result of the Average Lower Agglomeration Rupture Degree; ALARDopt, Brute Force result of Average Lower Agglomeration Rupture Degree; t(s) represents the running time in seconds. *Error* is the absolute gap, which is the magnitude of the difference between the values of ARD,

ALARD and the results of ARD, ALARD obtained by the proposed algorithm. Furthermore, 25%, 50%, 75% and 100% indicate the edge density of the graph G.

The proposed algorithm is implemented in JAVA and tested on i5-7600U machine with 2.9 GHz processor and 8 GB RAM. Clearly, we can see that the results of the actual ARD and ALARD is almost similar the result of ARD and ALARD obtained by proposed algorithm in Tables 2 and 3. We also tested our algorithm for the medium size graphs whose numbers of vertices more than *100*. Since we don't know actual values of ARD and ALARD, we give only heuristic result of ARD and ALARD with CPU time in the Tables 4 and 5. As a result, we have tested the algorithm on some graph families which are used in Theorems 1–6. Then, same values of ARD and ALARD are obtained as given Theorems 1–6.

	25%			50%			75%			95%		
|V|	ARD	ARDopt	Error	ARD	ARDopt	Error	ARD	ARDopt	Error	ARD	ARDopt	Error
10	0	0	0	0	0	0	1	1	0	0	0	0
11	1	1	0	1	1	0	0	0	0	0	0	0
12	0	0	0	0	0	0	-1	-1	0	0	0	0
13	0	0	0	1	1	0	1	1	0	0	0	0
14	-1	-1	0	-1	-1	0	0	0	0	0	0	0
15	2	2	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	-1	-1	0	0	0	0
17	2	2	0	0	0	0	0	0	0	0	0	0
18	-1	-1	0	-1	-1	0	0	0	0	0	0	0
19	-1	-1	0	-1	-1	0	-1	-1	0	0	0	0
20	-1	-1	0	0	0	0	-1	-1	0	0	0	0
21	-1	-1	0	-2	-2	0	0	0	0	0	0	0
22	-1	-1	0	-1	-1	0	-1	-1	0	0	0	0
23	1	1	0	-2	-2	0	-1	-1	0	0	0	0
24	-3	-3	0	-2	-2	0	0	0	0	0	0	0
25	-3	-3	0	-3	-3	0	0	0	0	0	0	0
26	-3	-3	0	-3	-3	0	1	1	0	0	0	0
27	-4	-4	0	-2	-2	0	-1	-1	0	0	0	0

Table 2: Computational experiments on small-sized graphs for ARD.

	25%			50%			75%			95%		
|V|	ALARD	ALARDopt	Error	ALARD	ALARDopt	Error	ALARD	ALARDopt	Error	ALARD	ALARDopt	Error
10	-11/10	-11/10	0	-14/10	14/10	0	-6/10	-6/10	0	-4/10	-4/10	0
11	-7/11	-6/11	-1/11	-14/11	-14/11	0	-12/11	-12/11	0	-6/11	-6/11	0
12	-9/12	-9/12	0	-28/12	-28/12	0	-32/12	-32/12	0	-6/12	-6/12	0
13	-19/13	-19/13	0	-19/13	-19/13	0	-18/13	-18/13	0	-8/13	-8/13	0
14	-32/14	-32/14	0	-34/14	-34/14	0	-26/14	-26/14	0	-10/14	-10/14	0
15	-15/15	-14/15	-1/15	-37/15	-37/15	0	-30/15	-30/15	0	-10/15	-10/15	0
16	-32/16	-32/16	0	-48/16	-48/16	0	-42/16	-42/16	0	-12/16	-12/16	0
17	-23/17	-23/17	0	-43/17	-43/17	0	-40/17	-40/17	0	-14/17	-14/17	0
18	-43/18	-43/18	0	-58/18	-58/18	0	-40/18	-40/18	0	-16/18	-16/18	0
19	-53/19	-53/19	0	-56/19	-56/19	0	-50/19	-50/19	0	-12/19	-12/18	0
20	-63/20	-63/20	0	-69/20	-69/20	0	-53/20	-53/20	0	-20/20	-20/20	0
21	-90/21	-87/21	-3/21	-99/21	-99/21	0	-70/21	-70/21	0	-14/21	-14/21	0
22	-86/22	-86/22	0	-105/22	-104/22	-1/22	-59/22	-59/22	0	-24/22	-24/22	0
23	-95/23	-91/23	-4/23	-109/23	-109/23	0	-62/23	-62/23	0	-26/23	-26/23	0
24	-133/24	-133/24	0	-132/24	-132/24	0	-82/24	-82/24	0	-28/24	-28/24	0
25	-143/25	-142/25	-1/25	-144/25	-144/25	0	-95/25	-95/25	0	-30/25	-30/25	0
26	-148/26	-148/26	0	-168/26	-166/26	-2/26	-96/26	-96/26	0	-32/26	-32/26	0
27	-157/27	-157/27	0	-167/27	-167/27	0	-105/27	-105/27	0	-36/27	-36/27	0

Table 3: Computational experiments on small-sized graphs for ALARD

	20%			30%			40%			50%		
|V|	ARD	ALARD	t(s)	ARD	ALARD	t(s)	ARD	ALARD	t(s)	ARD	ALARD	t(s)
100	-33	-4400/100	3.8	-36	-4630/100	3.6	-31	-4203/100	2.9	-26	-3712/100	3.2
110	-42	-5809/110	4.7	-42	-5788/110	4.3	-37	-5293/110	3.1	-33	-4580/110	3.0
120	-48	-7075/120	5.5	-48	-7022/120	5.2	-42	-6346/120	4.0	-35	-5565/120	3.4
130	-51	-8399/130	7.3	-50	-8379/130	6.6	-43	-7652/130	5.0	-33	-6610/130	7.7
140	-61	-9972/140	8.8	-59	-9966/140	8.4	-48	-8990/140	8.8	-39	-7793/140	5.9
150	-72	-12293/150	13.0	-65	-11686/150	10.1	-60	-10548/150	9.7	-46	-9003/150	6.6
160	-73	-13783/160	15.4	-69	-13504/160	12.5	-60	-12125/160	10.9	-47	-10418/160	6.7
170	-82	-16071/170	19.7	-71	-15485/170	18.3	-66	-13863/170	11.2	-53	-11775/170	7.8
180	-79	-18034/180	22.6	-72	-17357/180	18.0	-68	-15719/180	12.9	-57	-13406/180	8.9
190	-95	-20661/190	28.8	-88	-19728/190	23.8	-76	-17706/190	17.6	-57	-15129/190	11.8
200	-101	-23233/200	35.3	-96	-22132/200	31.2	-80	-19638/200	20.2	-66	-16858/200	13.0

Table 4: Computational experiments on medium-sized graphs.

	60%			70%			80%			90%		
|V|	ARD	ALARD	t(s)	ARD	ALARD	t(s)	ARD	ALARD	t(s)	ARD	ALARD	t(s)
100	-19	-3033/100	2.1	-11	-4630/100	1.8	-8	-1536/100	2.7	-2	-786/100	1.7
110	-22	-3733/110	3.0	-16	-2812/110	2.2	-10	-1890/110	1.6	-4	-936/110	1.7
120	-26	-4553/120	2.6	-18	-3425/120	2.5	-9	-2279/120	1.9	-3	-1133/120	1.3
130	-27	-5380/130	3.4	-21	-4042/130	2.9	-11	-2692/130	1.9	-3	-1319/130	1.9
140	-29	-6356/140	3.8	-20	-4764/140	2.8	-10	-3148/140	2.6	-3	-1577/140	1.6
150	-36	-7397/150	4.8	-23	-5547/150	3.1	-14	-3657/150	2.3	-4	-1808/150	1.6
160	-36	-8423/160	5.7	-26	-6390/160	3.8	-16	-4207/160	1.7	-5	-2104/160	2.7
170	-40	-9559/170	4.9	-26	-7228/170	3.9	-15	-4789/170	2.2	-7	-2354/170	2.4
180	-42	-10818/180	6.6	-30	-8204/180	4.2	-16	-5407/180	2.7	-4	-2677/180	2.3
190	-44	-12186/190	7.0	-34	-9202/190	4.7	-18	-6083/190	3.7	-7	-2965/190	2.7
200	-54	-13659/200	8.9	-36	-10267/200	5.9	-20	-6819/200	4.5	-7	-3318/200	4.0

Table 5: Computational experiments on medium-sized graphs.

6 Conclusion

In this paper, we considered agglomeration-based rupture degree in graphs. We define and investigate the agglomeration rupture degree $r^{agg}(G)$ and the average lower agglomeration rupture degree $r_{av}^{agg}(G)$, then these values have been computed for well-known families of graphs. Finally, we proposed a polynomial time heuristic algorithm to find the set of the lower agglomeration rupture degree $r_{v_k}^{agg}(G)$ for every vertex and also the values of $r^{agg}(G)$ and $r_{av}^{agg}(G)$ for any graph G. Then, we present the results of computational experiments on graphs with up to *200* vertices. The results show that the proposed heuristic algorithm efficiently computes the values of $r^{agg}(G)$ and $r_{av}^{agg}(G)$ of a given graph G. Developing of several heuristics for computing the other agglomeration-based graph parameters of graphs are the subjects of future work.

References

[1] E. Aslan, Weak-rupture degree of graphs, *Int. Journal of Foundations of Computer Science*. **27**, 6 (2016) 725–738. doi:10.1142/S0129054116500258 ⇒126

[2] E. Aslan, Edge-rupture degree of a graph, *Dynamics of Continuous, Discrete and Impulsive Systems Series B: Applications and Algorithms*. **22**, 2 (2015) 155–161. ⇒126

[3] A. Aytac, H. Aksu, Some results for the rupture degree, *International Journal of Foundations of Computer Science*. **24**, 8 (2013) 1329–1338. doi:10.1142/S0129054113500366 ⇒126

[4] A. Aytac, H. Aksu, The rupture degree of some graphs, *Mathematica Balkanica*. **24**, 1-2(2010) 85–101. ⇒126

[5] A. Aytac, Z.N. Odabas, Computing the rupture degree in composite graphs, *International Journal of Foundations of Computer Science*. **21**, 3 (2010) 311–319. doi:10.1142/S012905411000726X ⇒126

[6] A. Aytac, T. Turaci, Vertex vulnerability parameter of gear graphs, *International Journal of Foundations of Computer Science*. **22**, 5 (2011) 1187–1195. ⇒125

[7] V. Aytac, T. Turaci, Relationships between vertex attack tolerance and other vulnerability parameters, *RAIRO - Theoretical Informatics and Applications*. **51**, 1 (2017) 17–27. doi:10.1051/ita/2017005 ⇒125

[8] G. Bacak-Turan, E. Oz, Neighbor rupture degree of transformation graphs $G^{xy\text{-}}$, *International Journal of Foundations of Computer Science*. **28**, 4 (2017) 335–355. doi:10.1142/S0129054117500216 ⇒126

[9] K.S. Bagga, L.W. Beineke, W.D. Goddard, M.J. Lipman, R.E. Pippert, A survey of integrity, *Discrete Applied Mathematics*. **37-38** (1992) 13–28. doi:10.1016/0166-218X(92)90122-Q ⇒125

[10] Z.N. Berberler, H.İ. Yildirim, T. İltüzer, İ Tunç, Agglomeration-Based Node Importance Analysis in Wheel-Type Networks, *International Journal of Foundations of Computer Science*. **32**, 3 (2021) 26–288. doi:10.1142/S0129054121500210 ⇒127

[11] M. E. Berberler, Z. N. Berberler, Measuring the vulnerability in networks: a heuristic approach, *Ars Combinatoria*. **135** (2017) 3–15. ⇒126, 136

[12] V. Chvatal, Tough graphs and Hamiltonian circuits, *Discrete Mathematics*. **5**, 3 (1973) 215–228. doi:10.1016/0012-365X(73)90138-6 ⇒125

[13] M. Cozzens, D. Moazzami, S. Stueckle, The tenacity of a graph, *17th Int. Conf. Theory and Applications of Graphs*, Wiley, New York, 1995, pp. 1111–1122. ⇒125

[14] R. Durgut, H. Kutucu and T. Turacı, Global distribution center number of some graphs and an algorithm, *RAIRO-Operations Research*. **53**, 4 (2019) 1217–1227. doi:10.1051/ro/2018119 ⇒125

[15] R. Durgut, T. Turacı, H. Kutucu, A heuristic algorithm to find rupture degree in graphs, *Turk. J. Elec. Eng. & Comp Sci*. **27** (2019) 3433 – 3441. doi:10.3906/elk-1903-29 ⇒126, 136

[16] H. Frank, I.T. Frisch, Analysis and design of survivable networks, *IEEE Transactions on Communications Technology*, **18**, 5 (1970) 501–519. doi:10.1109/TCOM.1970.1090419 ⇒125

[17] M.A. Henning, Trees with Equal Average Domination and Independent Domination Numbers, *Ars Combinatoria*. **71** (2004) 305–318. ⇒125

[18] H. A. Jung, On maximal circuits infinite graphs, *Annals of Discrete Mathematics*. **3** (1978) 129–144. doi:10.1016/S0167-5060(08)70503-X ⇒125

[19] A. A. Kunt, Z. N. Berberler, Efficient identification of node importance based on agglomeration in cycle-related networks, *International Journal of Foundations of Computer Science*. **31**, 7 (2020) 969–978. doi:doi:10.1142/S0129054120500379 ⇒127

[20] A. Kirlangic, G. Bacak-Turan, On the rupture degree of a graph, *Neural Network World*. **22**, 1 (2012) 39–51. doi:10.14311/NNW.2012.22.003 ⇒126

[21] O.K. Kurkçu, E. Aslan, A comparison between edge neighbor rupture degree and edge scattering number in graphs, *International Journal of Foundations of Computer Science* **29**, 7 (2018) 1119–1142. doi:10.1142/S0129054118500247 ⇒126

[22] I. Mishkovski, M. Biey, L. Kocarev, Vulnerability of complex networks, *Communications in Nonlinear Science and Numerical Simulation*. **16**, 1 (2011) 341–349. doi:10.1016/j.cnsns.2010.03.018 ⇒125

[23] Y. Li, S. Zhang, X. Li, Rupture degree of graphs, *International Journal of Computer Mathematics*. **82**, 7 (2005) 793–803. doi:10.1080/00207160412331336062 ⇒125, 130, 132, 133, 134

[24] Y. Li, The rupture degree of trees. *International Journal of Computer Mathematics*. **85**, 11 (2008) 1629–1635. doi:10.1080/00207160701553367 ⇒126

[25] F. Li, Isolated rupture degree of trees and gear graphs, *Neural Network World*. **25**, 3 (2015) 287–300. doi:10.14311/NNW.2015.25.015 ⇒126

[26] F. Li, X. Li, Computing the rupture degrees of graphs. *7th International Symposium on Parallel Architectures, Algorithms and Networks*. 2004, pp. 368–373, Hong Kong, China. ⇒126

[27] Z.N. Odabas, A. Aytac, Rupture degree and middle graphs, *Comptes Rendus de Lacademie Bulgare des Sciences*. **65**, 3 (2010) 315–322. ⇒126

[28] Y. J. Tan, W. Jun, H. Z. Deng, Evaluation method for node importance based on node contraction in complex networks, Systems *Engineering-Theory & Practice*. **11.11** (2006) 79–83. ⇒127

[29] T. Turacı, A. Aytac, Combining the Concepts of Residual and Domination in Graphs, *Fundamenta Informaticae*. **166**, 4 (2019) 379–392. doi:10.3233/FI-2019-1806 ⇒125

[30] T. Turacı, On combining the methods of link residual and domination in networks, *Fundamenta Informaticae*. **174**, 1 (2020) 43–59. doi:10.3233/FI-2020-1930 ⇒125

[31] T. Turaci, E. Aslan, The average lower reinforcement number a graph, *RAIRO-Theor. Inf. Appl.* **50**, 2 (2016) 135–144. doi:doi:10.1051/ita/2016015 ⇒125

[32] T. Turaci, On the average lower bondage number a graph, *RAIRO-Operations Research*. **50**, 4-5 (2016) 1003–1012. doi:10.1051/ro/2015062 ⇒125

[33] B.D. West, *Introduction to Graph Theory*. 2nd ed. Urbana, NJ, USA: Prentice Hall, 2001. ⇒125

APPENDIX

```
Void_function Process(graph_parameter mainGraph, node_array_parameter
arrayNode) {
ruptures[] ← Ø
for i ← 0 to arrayNode's length {
    ruptures[i] ← Agglomeration(mainGraph, arrayNode[i]) }
Integer ARD ← largest value of array ruptures
Double ALARD ← sum of all ruptures array elements / arrayNode's length
}. # end function

Integer_function Agglomeration(graph_parameter mainGraph, node_parameter
Node) {
Graph ← mainGraph # Cloning the master graph to avoid corruptions.
neighbors ← neighboring nodes corresponding to each index.
aggCluster ← neighbors[Node] # Finding the neighbors of the node.
for i ← 0 to aggCluster's length { # aggCluster nodes find their neighbors.
    temp[i] ← neighbors[aggCluster[i]]
    for j ← 0 to temp's length {
        if temp[i] isn't equal to Node {
```

```
            add temp[j] to neighbors } } }

add Node to aggCluster
sort aggCluster by contents from largest to smallest
for i ← 0 to aggCluster's length { # Reset row and column.
      for j ← 0 to Graph's length {
            Graph[j][aggCluster[i]] ← 0
            Graph[aggCluster[i]][j] ← 0 } }
components[][] ← newly formed graph sets # add new graphs.
for i ← 0 to length of components { # deleting the neighborhood from the
components array.
      for j ← 0 to length of components[i] {
      if aggCluster contains components[i][j] {
            components[i][0] ← Ø } } }
for i ← 0 to length of components {
      if the length of components[i] is 0 {
            remove the i. variable from components } }

# Creating tagged component ↓
labeledComponents[][] ← Ø
labeledComponents ← components

# new tags ↓
Integer tag_number ← 0
for i ← 0 to length of labeledComponents {
      for j ← 0 to length of labeledComponents[i] {
            tag_number ← tag_number + 1
            labeledComponents[i][j] ← tag_number } }

# remove if empty ↓
for i ← 0 to length of labeledComponents {
      if the length of labeledComponents[i] is 0 {
            remove the i. variable from labeledComponents
            remove the i. variable from components } }

# create new graph ↓
Integer value ← Graph's length – aggCluster's length + 1
newGraph[value][value] ← Ø # newGraph is the matrix with value*value
length.
```

```
for i ← 0 to length of labeledComponents {
    for j ← 0 to length of labeledComponents[i] {
        if neighbors contain components[i][j] {
            newGraph[0][labeledComponents[I][j]] ← 1
            newGraph[labeledComponents[I][j]][0] ← 1 } } }
for i ← 0 to length of components {
    for j ← 0 to length of components[i] {
        for k ← j to length of components[i] {
            Integer a ← Graph[components[i][j]][components[i][k]]
            newGraph[labeledComponents[i][j]][labeledComponents[i][k]]← a
            Integer b ← Graph[components[i][k]][components[i][j]]
            newGraph[labeledComponents[i][k]][labeledComponents[i][j]] ← b
} } }

return function Rupture(newGraph) # Branched into the heuristic rupture
algorithm.
}. # end function
```

Received: April 27, 2023 • Revised: July 16, 2023

Acta Univ. Sapientiae Informatica **15,** 1 (2023) 146–169

DOI: 10.2478/ausi-2023-0011

Enhanced imagistic methodologies augmenting radiological image processing in interstitial lung diseases

József PALATKA
Doctoral School of Applied Informatics and Applied Mathematics
University Research, Innovation and Service Center
Óbuda University, Budapest, Hungary
email: jozsef.palatka@syneoshealth.com

Levente KOVÁCS
University Research, Innovation and Service Center
Óbuda University, Budapest, Hungary
email: kovacs@uni-obuda.hu ORCID: 0000-0002-3188-0800

László SZILÁGYI
Computational Intelligence Research Group,
Sapientia Hungarian University of Transylvania,
Târgu Mureș, Romania
University Research, Innovation and Service Center
Óbuda University, Budapest, Hungary
Department of Mechanical Engineering
University of Canterbury,
Christchurch, New Zealand
email: lalo@ms.sapientia.ro
szilagyi.laszlo@uni-obuda.hu ORCID: 0000-0001-6722-2642

Key words and phrases: interstitial lung diseases, high-resolution computed tomography, radiography image processing, complex networks, convolutional neural networks, computer-aided diagnostics, deep learning, machine learning

Abstract. Interstitial Lung Diseases (ILDs) represent a heterogeneous group of several rare diseases that are difficult to predict, diagnose and monitor. There are no predictive biomarkers for ILDs, clinical signs are similar to the ones for other lung diseases, the radiological features are not easy to recognize, and require manual radiologist review. Data-driven support for ILD prediction, diagnosis and disease-course monitoring are great unmet need. Numerous image processing techniques and computer-aided diagnostic and decision-making support methods have been developed over the recent years. The current review focuses on such solutions, discussing advancements on the fields of Quantitative CT, Complex Networks, and Convolutional Neural Networks.

1 Introduction

Interstitial lung diseases (ILDs) refer to a group of over 200 diverse disorders that involve inflammation and progressive fibrosis of lung interstitium, representing an important morbidity and mortality cause. The incidence of ILD ranges from 1 to 31.5 per 100,000 person-years and prevalence ranged from 6.3 to 71 per 100,000 people [20]. It is more common in the elderly population, median age at diagnosis is over 60 years [26].

ILD causes inflammation and scarring (fibrosis) of the interstitium, making the oxygen difficult to pass into the bloodstream. This can result in symptoms such as shortness of breath, cough, fatigue, and chest pain. ILD can also cause a decreased tolerance for physical activity, and in more advanced cases, can lead to respiratory failure; besides being seriously debilitating, it is significantly affecting the patients' quality of life.

There are many different types of ILD, with overlapping clinical, radiological, and pathological features. The most common type is idiopathic pulmonary fibrosis (IPF), which is of unknown cause. Other ones that are more prevalent are: connective tissue disease-associated ILD in people with autoimmune diseases (e.g. rheumatoid arthritis and scleroderma), hypersensitivity pneumonitis, sarcoidosis, and drug-induced ILD (DIILD).

ILD is known to be difficult to diagnose and treat, and management typically involves a multidisciplinary approach made of medications, oxygen therapy, pulmonary rehabilitation, and lung transplantation. Treatment efficacy is usually measured by changes in pulmonary function (forced vital capacity – FVC), more precisely the reduction of FVC decline over time, changes of exercise tolerance, or progression-free survival.

Beyond the individual patient burden, the economic impact of ILD is also significant, both on a personal and societal level. Here are some factors that contribute to the economic burden of ILD.

1. Healthcare costs: ILD often requires a suite of highly specific diagnostic tools, including radiographic imaging, lung function tests, bronchoscopy, and sometimes lung biopsy. For the progressive nature of the disease, ongoing monitoring, medication management, and specialist oversight is necessary. The costs associated with these medical services, including hospitalizations, medications, and regular follow-up visits, contribute to the economic burden. The home care for these patients is also substantial (assistance with daily activities, transportation to medical appointments, and emotional support).

2. Treatment expenses: ILD treatment may involve a combination of medications, such as corticosteroids, immunosuppressants, and antifibrotic drugs, depending on the specific type of ILD. These medications can be costly, and the duration of treatment may extend over a long period, further increasing the financial impact. In severe cases of ILD, where conservative treatment options have been exhausted, lung transplantation remains the only option. Lung transplantation is a complex and expensive procedure with substantial associated costs.

3. Lost productivity: ILD can significantly impact a person's ability to work and engage in daily activities. Consequently, individuals with ILD may experience decreased work hours, reduced productivity, or even complete disability, leading to income loss and further diminished quality of life.

The drug-induced interstitial lung disease is a specific type of ILDs that deserves a separate evaluation. This is a heterogenous group of pulmonary parenchymal diseases that occur in relation to exposure to certain drugs. To-date 1,653 drugs and procedures are associated with ILDs, and the list is increasing. Medicines used in several disease areas are on the list including many highly promising oncology products that are meant to cover areas of great unmet need [11]. DIILDs are an outcome of a medication administered to patients, hence keeping it at lowest possible incidence is a moral obligation for the physicians and drug-makers. Furthermore, DIILDs are darkening the results of certain drugs that are otherwise highly successful (e.g. trastuzumab deruxtecan (T-DXd), Enhertu®, the most successful oncology product discovered in the recent years [25]) and are seriously limiting their use and their therapeutic potential.

2 Predicting, diagnosing and monitoring ILDs

Despite the recent advancements of technology and medicinal science, the effective management of patients with ILDs is still insufficient at three main levels: the early detection of ILD, accurate prognostication using baseline data, and accurate and precise monitoring of disease response to therapy through high-resolution computer tomography (HRCT) [6].

The diagnosis of ILDs is based on integrated clinical evaluation, pulmonary function tests, radiological assessments, lab tests and, in some cases, histopathological examination that involves the collaboration of a large multidisciplinary team.

ILDs should be considered in the differential diagnosis of adults presenting with unexplained exertional shortness of breath, chronic cough, and/or crackles on chest auscultation, especially when the common pulmonary disorders can be ruled out. ILDs classically produce the "3Cs": cough, clubbing of the nails, and coarse crackles on auscultation [32]. At clinical evaluation a detailed medical history is obtained, including environmental and drug exposure history, full body examination with focus on clinical signs on the patients' hand, joint, and skin. A review of the patients' medication is needed in search of agents that are known to cause DIILDs. Common drugs associated with ILD are cancer therapies (i.e., bleomycin, immune checkpoint inhibitors), rheumatologic agents, amiodarone, and antibiotics (i.e., nitrofurantoin) and several others. A thorough family history focusing on idiopathic interstitial pneumonia and autoimmune disease should also be performed.

Pulmonary function tests (PFTs) are done to assess lung function and help determine the presence and severity of restrictive or obstructive lung disease. They typically include measurements of lung volumes, and of the maximum amount of air a person can forcefully exhale after taking a deep breath (forced vital capacity – FVC); as well as diffusing capacity for carbon monoxide (DLCO), and spirometry. Patients with ILDs typically exhibit reduced FVC, reduced total lung volume, and reduced diffusing capacities, though these values may appear normal early in the disease course, and when combined pulmonary fibrosis and emphysema is present [8].

High-resolution computed tomography (HRCT) is a key imaging modality for evaluating ILDs. HRCT scans provide detailed images of lung structures, allowing the detection of characteristic patterns associated with different ILD subtypes. Radiological features, such as ground-glass opacities, reticular or honeycomb patterns, nodules, and distribution patterns, help guide the diagnosis and classification of ILDs. Computer-aided evaluation of HRCT images,

in support of ILD diagnosis is a rapidly developing area, but the breakthrough has not yet been achieved [30].

Serological and immunological blood tests may be conducted to assess markers, autoantibodies, or immunological abnormalities that could indicate an underlying connective tissue disease associated with ILDs, such as rheumatoid arthritis, systemic sclerosis, or sarcoidosis.

Cytological, or histopathological examination may be required to establish a definitive diagnosis and determine the underlying histopathological features. Bronchoalveolar lavage material or lung tissue obtained through biopsy helps identify the characteristics of the interstitial inflammation, fibrosis, or other specific changes, aiding in the classification of ILDs. Recently whole transcriptome RNA sequencing of the biopsy tissue sample was found successful in classifying ILDs. Gene expression analyses can help to distinguish between types of ILDs. These are areas of intensive ongoing research [26].

2.1 Clinical challenges

The prevalence of ILDs is low, and this already makes it difficult to be recognized. Their clinical features (such as shortness of breath, cough, and restrictive lung function patterns) are similar to those seen in common lung diseases, therefore the early diagnosis is a challenge. Some patients present for evaluation of cough and dyspnea several years before being diagnosed with ILD, after receiving initial diagnoses of chronic obstructive pulmonary disease, heart failure or other diseases.

Besides of the low prevalence, ILDs have no known clinical or radiological predictive biomarkers, only risk factors of low specificity associated with this disease have been identified (age, male sex, cigarette smoking, hepatitis C infection, history of tuberculosis, history of pneumonia, COPD, exposure to toxic substances [13]). Unless there is a specific suspicion for ILDs, the diagnosis can be easily overlooked. The incorrect or delayed diagnosis leads to worsening of the disease and the use of invasive and/or costly diagnostic procedures (like biopsies) of questionable value.

Accurate diagnosis and classification of ILDs often require input from various medical specialists, including pulmonologists, radiologists, pathologists, and rheumatologists. Coordinating and integrating the expertise of multiple disciplines is not easy, particularly in regions with limited access to specialized medical resources. Management of ILDs and care for patients with ILDs remain a challenge throughout the course of their disease for lack of disease-modifier treatment options.

2.2 Radiological challenges

The role of HRCT is critical for diagnosing ILDs. The different types of ILDs express specific imaging features like reticulation, consolidation, micronodules, emphysema, honeycombing, ground-glass opacity and a combination of these, that are essential for diagnosis. Interpreting radiological findings can be challenging as there is substantial inter-observer variability even between experienced radiologists, and the imaging patterns often are mixed, and the features observed overlap among different ILD subtypes. On the other hand, visual evaluation of ILD by HRCT has little sensitivity to objective changes in disease severity over short follow-up periods.

Image standardization is difficult to achieve, for normal lung tissue idiosyncrasy and artifacts, caused by patient movement during scanning, different types of breathing. Though HRCT is the accepted standard to be used when ILD is suspected, there are inconsistencies between CT technical characteristics, different scanner manufacturers, models, acquisition protocols, and reconstruction algorithms. International collaborations would be very important between Academia, Pharma Industry and Healthcare to develop comprehensive guidelines for imaging standards and basic image-processing algorithms.

The most clinically meaningful information hiding within the very large medical imaging datasets is generally unstructured, and require extensive preprocessing (including segmentation, filtering, registration, and labeling) before further analysis can occur. Defining the segment of interest for evaluation requires manual or semi-manual annotation.

2.3 Image processing challenges

Image standardization is difficult to achieve, for normal lung tissue idiosyncrasy, and artifacts caused by patient movement during scanning, different types of breathing. Though HRCT is the accepted standard to be used when ILD is suspected, there are inconsistencies between CT technical characteristics, different scanner manufacturers, models, acquisition protocols, and reconstruction algorithms. International collaborations would be very important between Academia, Pharmacological Industry and Healthcare to develop comprehensive guidelines for imaging standards and image-processing algorithms.

The most clinically meaningful information hiding within the very large medical imaging datasets is generally unstructured, and require extensive preprocessing (including filtering, registration, and labeling) before further analysis can occur. Defining the segment of interest for evaluation requires manual

or semi-manual annotation, therefore until the machine-learning mechanisms reach their advanced stage of development, human pre-processing remains essential.

The algorithms developed for image recognition still need to improve for the precise classification of the patterns seen (honeycombing, reticulation and ground glass opacity, etc.), especially when they appear mixed on the images studied. Multiple methods have been proposed for computer-aided object recognition and classifying (multi-scale rotation invariant algorithms with eg. Gabor filter, patch-based image representation methods and others), the optimal tool is yet to be found.

There is a pronounced need for a computer-based tool that operates on data from radiological images and clinical data, that predicts ILDs and/or reliably detects them at earlies (even subclinical) stages of the disease, and enables monitoring of disease response to therapy.

3 Digital techniques for ILD diagnosis and monitoring

There is a pronounced need for computer-based tools that operate on data from radiological images and clinical domain, that predict ILDs and/or reliably detect them at earlier (even subclinical) stages of the disease and enables their longitudinal follow up and assessing the treatment outcomes.

A few decades ago only simple image analytics were used for image processing purposes. The novel biomarkers based on radiography images have only started to become available recently, their numbers are steadily increasing with the implementation of complex image analysis based on machine learning techniques.

3.1 Quantitative CT

Quantitative CT (QCT) provides an alternative to the visual evaluation, that is objective and reproducible by the use of computer-based techniques to analyze HRCT images. This method is based on simple statistical analysis of CT attenuation values of each targeted pixel of the lung images, without studying the correlation between them [4].

The only method commercialized and widely used to-date to quantify the pulmonary tissue is CALIPER (Computer Aided Lung Informatics for Pathology Evaluation and Rating), developed by Mayo Clinic of USA. The im-

age quantification done by CALIPER is based on histogram signature mapping techniques trained through datasets confirmed by expert radiologists. As part of the development for ILDs, the local histograms computed from the $15 \times 15 \times 15$ neighborhood of each of the parenchymal voxel were compared against the histogram of exemplars identified in the training phase, divided in 5 classes (emphysema, ground glass opacity, honeycombing, reticular infiltrates and normal tissue). Quantitative discriminability of a number of pairwise dissimilarity metrics based on the volume of interest histograms was examined using multi-dimensional scaling. Of several techniques Cramer Von Mises Distance was found to be most consistent with the expert grouping. CALIPER is easy to use and provides good support for ILD diagnosis and disease course monitoring, its performance however leaves room for improvement. The correlation of CALIPER results with physiologic parameters was generally strong but the correlation with the radiologist assessment of disease type and severity was only around 50%, hence can only be used in the context of clinical data [7].

3.2 Complex networks

The new methods of computer aided diagnosis (CAD) for lung HRCTs only provide a static evaluation of the images and require extensive computing skills and infrastructure. In response to this challenge, a novel technique was developed by Truşculescu et al. [31], built on a CN analytic approach for imagistic aided diagnosis fitness for the possibility of achieving relevant data for ILD management.

The method was developed on HRCT images from 65 patients with ILD and 31 with normal lung, acquired from Clinical Hospital of Infectious Diseases and Pneumophysiology Dr. Victor Babeș of Timisoara, Romania. Regions of interest were marked by a radiologist with high experience in imagistic diagnosing of ILDs. Three non-overlapping separate bands of Hounsfield Units (HU) have been created in line with the categories of the characteristic attenuations of the lung alterations. The images were then transformed into complex networks according to specific predefined attachment rules, based on the HU values of each pixel. Network nodes and connections have been defined based on the similarities in HU values of neighboring pixels. CN measurements were done for interconnectedness and size. Maximum degree number, total degree count and average degree count were evaluated.

The method was successful for early disease detection in one of the three bands (the one corresponding to ground glass opacity), partially successful in

the other (reticulation) and not successful in the third (emphysema). When used to assess disease course on sequential image sets for the same patient, the method was highly successful by showing close correlation with the changes of the clinical parameters [31].

3.3 Convolutional neural network-based methods

Substantial progress has been made in image recognition, with the advent of CNN-based solutions following the availability of large-scale annotated datasets like ImageNet which offered very comprehensive database of more than 1.2 million categorized natural images of 1000+ classes [15]. Obtaining datasets as comprehensively annotated as ImageNet in the medical imaging domain remains a challenge however, as data acquisition is difficult, and quality annotation is costly.

The implementation of machine-learning-based image analysis in the clinical management of ILDs requires extensive data sets for training purposes. Given the rarity of ILD access to high-quality medical images and clinical data is costly and difficult.

Tables 1 and 2 summarize the results achieved by a selection of CNN-based methods deployed in ILD prediction and diagnosis, presenting details of the application, information on the used data volume, and the main performance benchmark values claimed by the authors.

The solutions developed for detecting ILD patterns are broadly divided into two categories: patch-based methods and slice-based methods, with the desire to trend towards the latter, which allows a more generalized processing of images, without the tedious manual work of the patch-based techniques.

3.3.1 Patch-based methods

A plethora of published works refer to patch-based classification of ILD patterns, after manual extraction of patches by radiologists [29]. Informative features are extracted from several ILD patches with the help of different feature extraction techniques for the classification of ILD patterns. The selection of an appropriate classifier is very important. Common methods used are k-nearest neighbors [24], artificial neural network, and support vector machines [19].

The classification accuracy of these methods steadily increased over time. An early method involving near-affine-invariant texture-based feature descriptor based on wavelet transformation used to classify the five ILD patterns (healthy, emphysema, GGO, fibrosis, and micronodules) showed a classifica-

Paper	Population	Data volume	Clinical application	Method	Results
Kim et al [21]	ILD	1200 ROIs	ILD features identification	Radiologist-identified ROIs, paired with a CNN classifier	$ACC = 81.27 - 95.12\%$
Christe et al 2019 [14]	IPF	105 cases	ILD features classification	CAD system using a CNN trained by radiologists	$F1 = 56\%$, $ACC = 81\%$
Agarwala et al [1]	ILD	1946 ROIs	ILD features identification	Radiologist-identified ROIs, paired with a CNN classifier, using faster R-CNN based detector network	$F1 = 55 - 88\%$
Huang et al [19]	ILD	unknown	ILD features classification	Unsupervised deep CNN	$F1 = 97.91\%$
Bermejo-Peláez et al [9]	ILD	37,424 ROIs	Automated identification and classification of ILDs	multi-model ensemble of deep CNN	Average $TPR = 91.41\%$, $TNR = 98.18\%$
Anthimopoulos et al [2]	ILD	14,696 ROIs	Lung pattern classification	Deep CNN	$F1 = 85.47\%$
Anthimopoulos et al [3]	ILD	172 scans	Semantic segmentation of ILD patterns	CNN, dilated convolution	$ACC = 81.8\%$

Table 1: Relevant studies in ILD diagnosis

Paper	Population	Data volume	Clinical application	Method	Results
Choi et al [12]	IPF	516 cases, 500 image montages each	Correlation with progressive pulmonary fibrosis	Diagnostic probability of UIP, with a deep CNN	hazard ratio 1.73 (CI95%: 1.40-2.14, $p < 10^{-4}$)
Budzykowski et al [10]	IPF	169 patients, 6 ROIs each	Association with genetic biomarkers (TOLLIP, MUC5B), survival	Extraction of radiomic features, paired with a CNN classifier	nine first-order texture features and one fractal feature were correlated with TOLLIP-1 mutations (AUC: 0.54 to 0.74); five Laws' filter features were correlated with TOLLIP-2 mutations (AUC: 0.53 to 0.70)
Park et al [23]	IPF	193 patients	Analysis of predictive factors for a decline in forced vital capacity (FVC)	Texture-based automated system used in-house software to quantify six ILD imaging features vs changes in FVC values	Reticular Opacity (RO) is sole independent predictor for FVC decline (p = 0.012; adjusted odds ratio, 1.047). ROC for RO was 0.641, optimal RO cutoff value was 22.05% (sensitivity, 50.0%; specificity, 81.4%; negative predictive value, 89.1%).

Table 2: Relevant studies in ILD prognostication

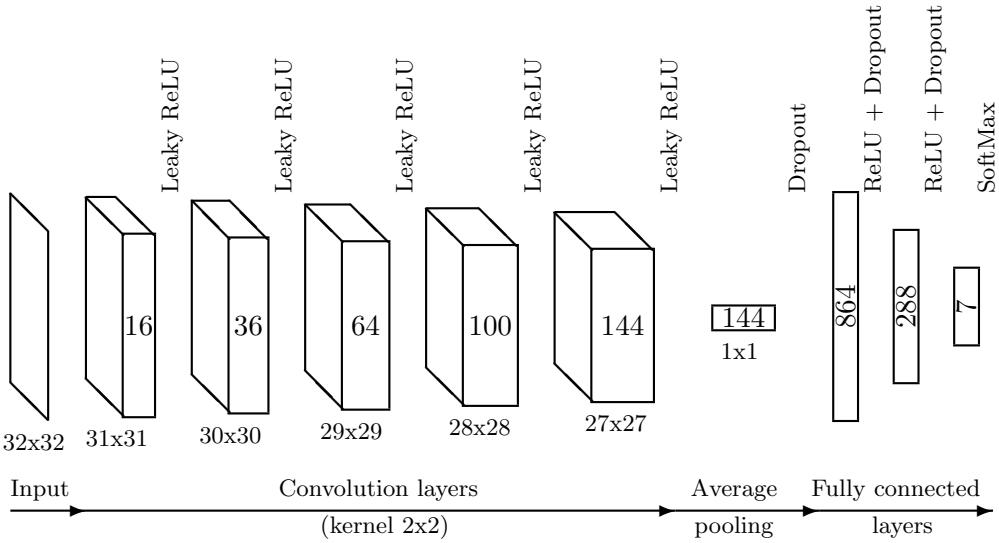


Figure 1: The structure of the CNN used for patch-based recognition of ILD, adapted from [2].

tion accuracy of 76.9% [16]. Another method based on texture and gradient features for patch-based classification of the ILD patterns using support vector machine, reported F1-scores ($F1 = 2 \times (\text{precision} \times \text{recall})/(\text{precision} + \text{recall})$) for healthy, emphysema, GGO, fibrosis, and micronodules are 84%, 75.3%, 78.2%, 84.1%, and 85.7%, respectively [28].

The first deep CNN designed for lung pattern classification achieved an average F1-score of 85.47% across 7 classes of CT image patches (6 typical ILD patterns and healthy tissue). The network, shown in Figure 1, was built of 5 convolutional layers, each of them used kernels of 2×2 and Leaky ReLU as activation function. The method used three dense layers with 7 outputs, in line with the ILD image classes targeted: ground glass opacity, micronodules, consolidation, reticulation, honeycombing and a combination of GGO/reticulation and healthy tissue. Training was done on a dataset of 14,696 image patches extracted from 120 HRCT images obtained from healthcare institutions [2].

More recently, another such method reported F1-score of 97.91%. This was achieved with a deep CNN architecture built from six different convolutional layers followed by batch normalization layers and ending with a fully connected layer. The network used input patches of size 32×32 extracted from ILD HRCT images. Each layer worked with kernel size of 2×2, with number of

kernels gradually increasing layer to layer from 32 to 192. ReLU was employed as activation function. The learning of unlabeled data was done through an unsupervised method. The results showed that the proposed CNN architecture outperforms most of the state-of-the-art ones [19].

Another promising method has been recently reported to identify radiographic patterns that precede the development of ILD with an average sensitivity of 91.41% and average specificity of 98.18% across 8 classes of HRCT pattern (healthy tissue, five interstitial features subtypes and two emphysematous classes), on 37,424 radiographic tissues extracted from 208 CT images. Deep learning approach was used on a highly complex ensemble of CNN architecture that comprises three different architectures such as 2D, 2.5D, and 3D for the classification of ILD abnormalities. Each individual network was trained from scratch from the database, the outputs of the networks are summed up in a weighted manner and combined to form the overall output of the ensemble. The resulting ensemble achieved a higher performance compared to each of the individual models, and the reported CNN methods of the domain, showing the potential of combined use of a suite of classifiers [9]. The network architectures involved in this study are depicted in Figure 2.

The results achieved with patch-based classification methods are remarkable, however their use is limited by computational challenges, the manual annotation involved and their limitation to be used for screening of HRCT patterns at slice level.

3.3.2 Slice-based methods

Early attempts to classify HRCT slices depending on the presence of pathology used pretrained AlexNet, but reported poor classification results. More complex solutions testing multiple systems (Cifarnet, AlexNet, GoogLeNet) showed improved slice level classification accuracy of ILD patterns on HRCT slices, the highest F1-Score achieved being with GoogLeNet, of 92% [27]. Details of the deployed network structures are given in Figure 3.

Deep CNN network with dilated filters were reported to be successful to segmentation of ILD patterns. The network proposed used input images of any arbitrary size of lung HRCT and the generated outputs were label maps. The network, as shown in Figure 4, consisted of eight convolutional layers having different dilation rates that increase exponentially. This helped to increase the receptive field while linearly growing the number of parameters. The performance was evaluated on 172 HRCT slices collected from two hospitals such

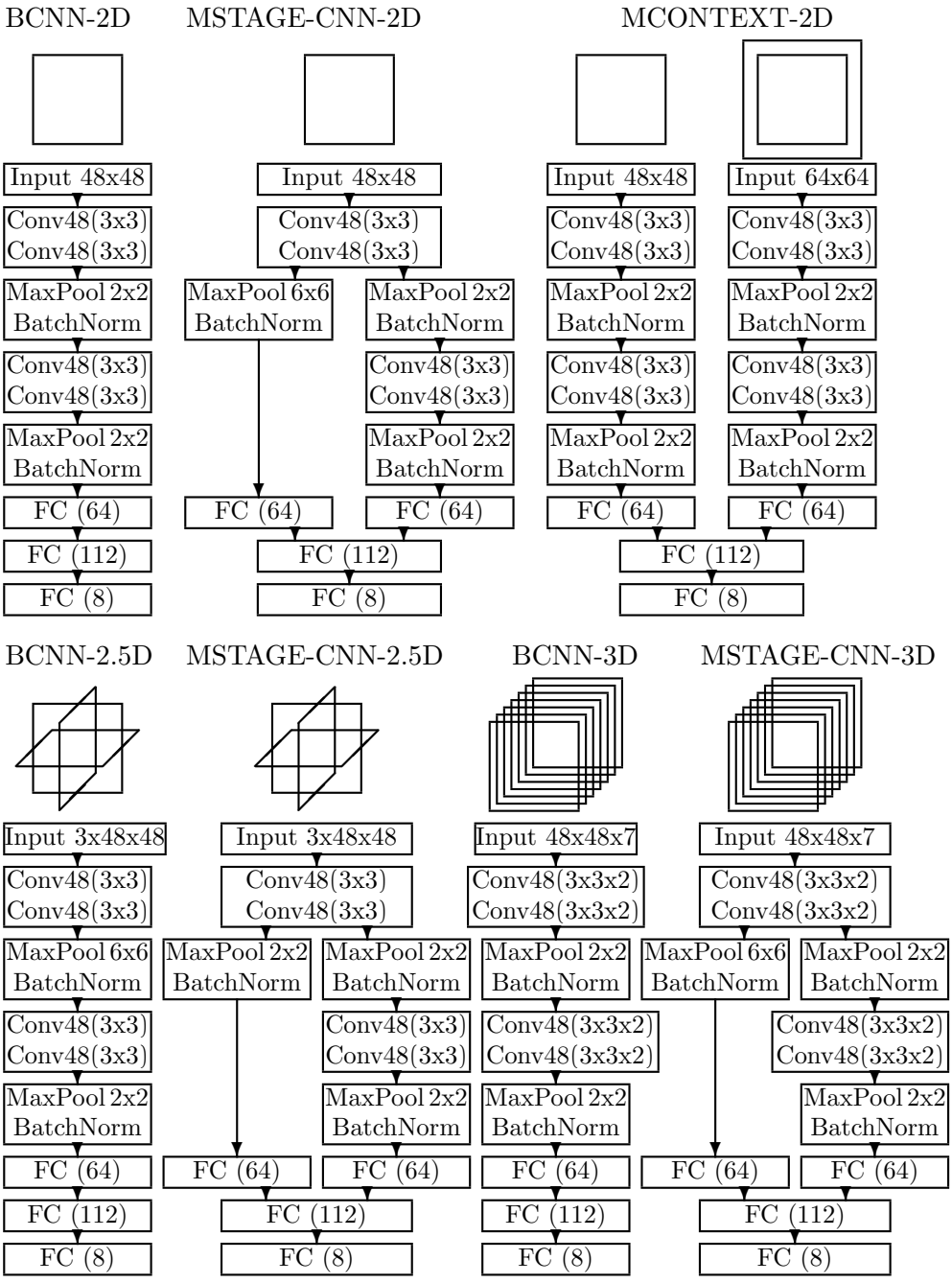


Figure 2: The CNN structures proposed by Bermejo-Peláez et al. [9].

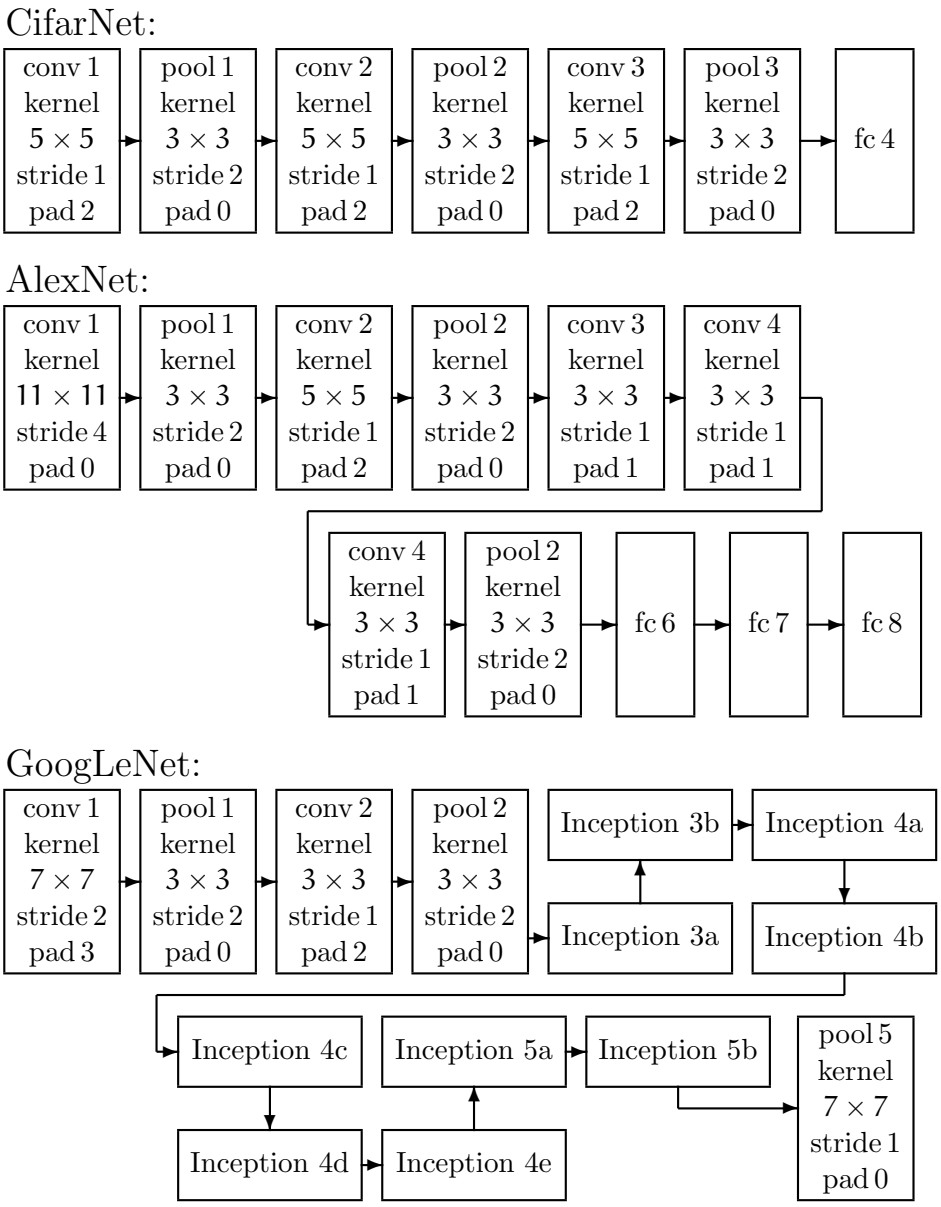


Figure 3: CNN architectures proposed by Shin et al, adapted from [27]. Each Inception module contains six convolution layers, a pooling layer and a concatenate layer.

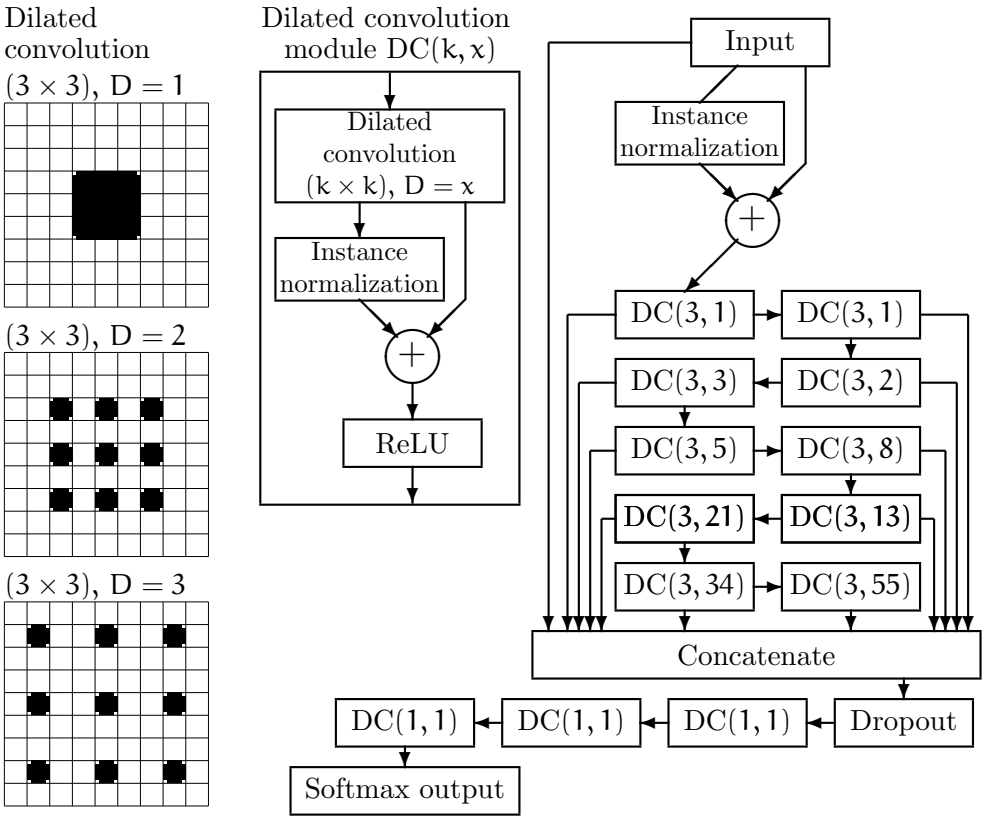


Figure 4: The CNN structure proposed by Anthimopoulos et al, adapted from [3]

as Geneva University Hospital and Bern University Hospital. This network achieved an accuracy of 81.8% [3].

Agarwala et al. [1] proposed a solution for localization of the typical ILD patterns in a HRCT slice using a more efficient region-based convolutional network (R-CNN) driven object detection network. GoogLeNet architecture has been modified for lower complexity by using only 5 inception blocks instead of 9 and used to extract the image features for faster object detection. The features provided by the fifth inception block were used as proposal for finding the targeted region. To overcome the limited amount of annotated training data, data augmentation techniques (flip, rotation, change of contrast, and addition of Gaussian noise) have been used. Six ILD patterns have been used (Consolidation, Emphysema, Fibrosis, GGO, Micronodule and healthy tissue).

The above-oultined method performed very well across all patterns, little less for fibrosis, achieving F1-scores between 0.55 and 0.86. It is fast and accurate, avoids the hassles of lung field segmentation and could be used in the screening of ILD using HRCT image slices.

4 Public datasets of ILD-related images

Beyond the private datasets of Healthcare institutions, there are multiple data sources and registries for ILD patients that provide access to radiography images (mostly HRCT) and clinical data points for research purposes. The most widely used open data sources are discussed below.

4.1 Lung Tissue Research Consortium database

The Lung Tissue Research Consortium database (LTRC-DB[1]) was a resource program of the National Heart, Lung, and Blood Institute that provides CT scans, as well as biospecimens to researchers of the domain. The LTRC was established in 2005 by the National Institutes of Health based on a coalition of 4 major clinical centers from the USA: Mayo Clinic Rochester, University of Michigan–Ann Arbor, University of Pittsburgh, and Temple University. During its active years between 2005 to 2019, the LTRC's main task was to collect, store, and make available imaging samples and clinical data from patients with various types of lung diseases. The LTRC sample and data set was sourced from more than 4,200 patients, with over 100 of cases with one of the several forms of ILDs.

4.2 Multimedia database of Interstitial Lung Diseases

The multimedia database for ILDs (MD-ILD[2]) was developed as part of the Talisman project at the University Hospital of Geneva and is made publicly available. This highly valuable database is specific to ILDs, and contains standard HRCT image series of 10-mm slice spacing. Annotations of pathological lung segments together with clinical parameters from patients with pathologically proven diagnoses of ILDs have been obtained from expert radiologists and treating clinicians. There are 128 ILD cases in the database with one of the 13 histological diagnoses of ILDs, 108 image series are available with

[1] https://ltrcpublic.com
[2] http://medgift.hevs.ch/wordpress/databases/ild-database/

1,946 delineated polygons of annotated lung parenchyma patterns, along a comprehensive dataset of 99 clinical parameters related to ILDs.

4.3 Inselspital Interstitial Lung Diseases database

The Inselspital ILD database (INSEL-DB) is a smaller but very well structured library, which has two main parts, based on the accompanying annotations. The first part consists of 60 HRCT image series of 9-mm slice spacing lung scans with annotated tissue from 2 radiologists (INSEL-DB-Seg), and the second consists of 105 HRCT lung scans provided by the ILD board of Bern University Hospital. The HRCT scans have been collected retrospectively, between October 2015 and June 2017. Demographic, clinical and laboratory data for each patient (e.g. sex, age, smoking history, duration of illness, lung function tests, results of blood tests) was also collected and made available to researchers.

4.4 AIFPR: Australian Idiopathic Pulmonary Fibrosis registry

A large regional library of patients data has been developed in Australia by University of Sydney[3]. AIPFR has been developed in Australia by University of Sydney3 with IPF cases recorded between February 2011 and December 2020. A total of 21 sites participated across Australia and New Zealand collecting impressively large datasets from over 2,700 patients enrolled in the program. Longitudinal follow-up data was also collected every 6 months when possible. The dataset collected includes clinical parameters (PFT), patient reported outcome (PRO) data, HRCT images and blood samples data. Data collection finished on 31 December 2021, the data continues to be available for research.

4.5 Open Source Imaging Consortium (OSIC) data repository

OSIC, an international group of leading experts, established OSIC Data Repository[4] on 22nd of May, 2019. This global, not-for-profit organization is a cooperative and open-source effort between academia, healthcare and industry to enable rapid advances in the detection and diagnosis of these conditions through digital imaging and machine learning.

[3]https://www.sydney.edu.au/medicine-health/our-research/research-centres/aildr.html
[4]https://www.osicild.org/

OSIC was created to drive collaboration between distant partners and to unite their capabilities. It was created to use artificial intelligence and other technological advances to build, and learn from, the largest and most diverse image and clinical database for fibrotic lung diseases. To-date OSCI has over 15,000 anonymized scans with accompanying clinical data, with over 106,000 anonymized data points from 1,843 patients with various forms of ILD. The enrollment of participants is ongoing.

5 Discussion

Over the recent years a myriad of computational techniques emerged in the support of radiographic image processing for ILDs. The example solutions discussed in the earlier sections outline encouraging trends, but the breakthrough has not yet been achieved. Many of the developed tools are highly performing, the competition for finding the optimal method is advanced and still ongoing. An ultimate breakthrough might not be possible however without addressing the key needs in image standardization, harmonization of definitions and classifications. This is not uncommon in the field of digital image-processing. For example, the automatic segmentation of brain tumors based on MRI data has been a widely researched topic for at least three decades. The development of segmentation methodology in the beginnings was similar to the current case of ILD: research groups were elaborating methods and techniques based on own data collections, usually captured by a low number of medical devices. Consequently their methods may have learned specific features of tumors together with parameters of the imaging equipment. Further on, the accuracy benchmark values were not objectively comparable with each other, because not even the testing data differed from team to team, but also the goals of the segmentation. Some were using single channel MRI volumes or only slices (cross sections), others lately turned to multi-spectral volumetric MRI data [17]. It is also necessary to mention, that this field grew together with the spectacular advances in the available computation speed and dynamic storage space, and the latest revolution of artificial intelligence brought by CNNs and deep learning. But the greatest factor in the development of brain tumor segmentation techniques represents the Brain Tumor Segmentation (BraTS) challenge [22, 5] organized yearly since 2012 by the Medical Image Computation and Computer-Assisted Interventions (MICCAI[5]) conference, which provided the common data, common goals and a common evaluation framework. BraTS

[5] https://www.miccai.org

thus provoked the explosion in the field that is directly responsible for thousands of methods and works elaborated in this field. The amount of training and testing data, and also the variability of data sources widened year by year, and new questions or secondary goals were defined (e.g. give an estimate of the time the patient lived after the MRI data were recorded). The initial training dataset contained only 30 records, and they were not even formatted to the same volume size. The experience accumulated year by year made the BraTS challenge an easily accessible research for all, and this led to the exponential growth of developed methods. The whole arsenal of artificial intelligence got involved in various solutions not only in the direct classification of pixels, superpixels or patches, but also in the preprocessing of the data and postprocessing of the classification outcome to optimize the accuracy of the final result [18].

The history of BraTS could be considered a guideline in the field of automatic processing of ILD-related image and medical data. An open challenge could bring considerable advancement in the development of segmentation methods. The most appropriate goal to set could be the automatic segmentation (localization and quantification) of fibrotic tissues from series of chest CT scans that come from the same patient during observation time, and eventually to give some prognosis of the illness using further available medical data. As it was already stated, the organization of this challenge would require establishing a collection of ILD patient records, image data collected from multiple institutions and various CT scanners, each record accompanied by the same set of medical parameters, and a ground truth established by competent human experts. Building up these foundations could help the scientific community in understanding the background of ILDs and could cause spectacular advances in the methodology of ILD treatment.

6 Conclusion

Research on imaging biomarkers in ILD is advancing rapidly. Machine learning stands at the core of this process, supported by on deep-learning-based image analysis. Several clinical challenges could be addressed by this technology like the prediction, early detection and precise categorisation of ILDs, along the improved monitoring of the disease's natural course and response to therapy. The results seen with Quantitative CT, Complex Networks, and Convolutional Neural Networks hold the promise of a brighter future.

The accuracy of recognizing ILD features in HRCT images already exceeds 90% in some of the methods, however, the most precise techniques are still experimental and need advanced computational resources and substantial manual work for training and annotations. In an ideal world the image recognition techniques should be integrated onto the everyday Radiology and Pulmonology practice to operate on entire CT slices, without any specific pre-work from the radiologists. There are still unmet needs both to increase sensitivity and specificity of the methods, as well as to achieve solutions that run seamlessly on regular healthcare IT infrastructure. The integration of rapidly evolving digital biomarkers with the physiological, proteomic, and genomic data for patients will offer the greatest patient benefit.

Acknowledgements

This study was supported in part by the Consolidator Researcher Program of Óbuda University.

References

[1] S. Agarwala, A. Kumar, A.K. Dhara, S.B. Thakur, A. Sadhu, D. Nandi, Special Convolutional Neural Network for Identification and and Positioning of Interstitial Lung Disease Patterns in Computed Tomography Images, *Pattern Recognition and Image Analysis* **31**, 4 (2021) 730-738. ⇒155, 161

[2] M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe, S. Mougiakakou Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network, *IEEE Transactions on Medical Imaging* **35**, 5 (2016) 1207-1216. ⇒155, 157

[3] M. Anthimopoulos, S. Christodoulidis, L. Ebner, T. Geiser, A. Christe, S. Mougiakakou, Semantic Segmentation of Pathological Lung Tissue with Dilated Fully Convolutional Networks, *IEEE Journal of Biomedical and Health Informatics* **23**, 2 (2018) 714-722. ⇒155, 161

[4] S.Y. Ash, R. Harmouche, D.L. Lopez Vallejo, J.A. Villalba, K. Ostridge, R. Gunville, C.E. Come, J.O. Onieva, J.C. Ross, G.M. Hunninghake, S.Y. El-Chemaly, T.J. Doyle, P. Nardelli, G.V. Sanchez-Ferrero, H.J. Goldberg, I.O. Rosas, R. San Jose Estepar, G.R. Washko, Densitometric and local histogram based analysis of computed tomography images in patients with idiopathic pulmonary fibrosis, *Respiratory Research* **18** (2017) 45. ⇒152

[5] S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, M. Prastawa, et al., Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge, arXiv (2019) 1181.02629v2. ⇒164

[6] H. Barnes, S.M. Humphries, P.M. George, D. Assayag, I. Glaspole, J.A. Mackintosh, T.J. Corte, M. Glassberg, K.A. Johannson, L. Calandriello, F. Felder, A. Wells, S. Walsh, Digital Technology and the Future of Interstitial Lung Diseases, *Lancet Digit Health* **5** (2023) e41-50. ⇒149

[7] B.J. Bartholmai, S. Raghunath, R.A. Karwoski, T. Moua, S. Rajagopalan, F. Maldonado, P.A. Decker, R.A. Robb, Quantitative CT Imaging of Interstitial Lung Diseases, *Journal of Thoracic Imaging* **28**, 5 (2013) 298-307. ⇒153

[8] K. Berger, R.J. Kaner, Diagnosis and Pharmacologic Management of Fibrotic Interstitial Lung Disease, *Life (Basel)* **13**, 3 (2023) 599. ⇒149

[9] D. Bermejo-Peláez, S.Y. Ash, G.R. Washko, R. San José Estépar, M.J. Ledesma-Carbayo, Classifcation of Interstitial Lung Abnormality Patterns with an Ensemble of Deep Convolutional Neural Networks, *Scientific Reports* **10** (2020) 338. ⇒155, 158, 159

[10] J.D. Budzikowski, J.J. Foy, A.A. Rashid, J.H. Chung, I. Noth, S.G. Armato, Radiomics-based assessment of idiopathic pulmonary fibrosis is associated with genetic mutations and patient survival, *Journal of Medical Imaging* **8**, 5 (2021) 031903. ⇒156

[11] P. Camus, https://www.pneumotox.com, The drug-induced respiratory disease website, last accessed 15 June 2023. ⇒148

[12] B. Choi, S.Y. Ash, Deep Learning–based Classification of Fibrotic Lung Disease: Can Computer Vision See the Future?, *American Journal of Respiratory and Critical Care Medicine* **206**, 7 (2022) 812-814. ⇒156

[13] W.I. Choi, S. Dauti, H.J. Kim, S.H. Park, J.S. Park, C.W. Lee, Risk factors for interstitial lung disease: a 9-year Nationwide population-based study, *BMC Pulmonary Medicine* **18** (2018) 96. ⇒150

[14] A. Christe, A.A. Peters, D. Drakopoulos, J.T. Heverhagen, T. Geiser, T. Stathopoulou, S. Christodoulidis, M. Anthimopoulos, S. Mougiakakou, G. Stavroula, L. Ebner, Computer-Aided Diagnosis of Pulmonary Fibrosis Using Deep Learning and CT Images, *Investigative Radiology* **54**, 10 (2019) 627-632. ⇒155

[15] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009) pp. 248-255. ⇒154

[16] A. Depeursinge, D. Van de Ville, A. Platon, A. Geissbuhler, P.A. Poletti, H. Müller, Near-affine-invariant texture learning for lung tissue analysis using isotropic wavelet frames, *IEEE Transactions on Information Technology in Biomedicine* **16**, 4 (2012) 665-675. ⇒157

[17] N. Gordillo, E. Montseny, P. Sobrevilla, State of the art survey on MRI brain tumor segmentation, *Magnetic Resonance Imaging* **31** (2013) 1426-1438. ⇒164

[18] Á. Győrfi, L. Szilágyi, L. Kovács, A fully automatic procedure for brain tumor segmentation from multi-spectral MRI records using ensemble learning and atlas-based data enhancement, *Applied Sciences* **11** (2021) 564. ⇒165

[19] S. Huang, F.F. Lee, R. Miao, Q. Si, C. Lu, Q. Chen, A deep convolutional neural network architecture for interstitial lung disease pattern classification, *Medical & Biological Engineering & Computing* **58** (2020) 725-737. ⇒154, 155, 158

[20] B. Kaul, V. Cottin, H.K. Collard, C. Valenzuela, Variability in Global Prevalence of Interstitial Lung Disease, *Frontiers in Medicine* **8** (2021) 751181. ⇒147

[21] G.B. Kim, K.H. Jung, Y. Lee, H.J. Kim, N. Kim, S. Jun, J.B. Seo, D.A. Lynch, Comparison of Shallow and Deep Learning Methods on Classifying the Regional Pattern of Diffuse Lung Disease, *Journal of Digital Imaging* **31** (2018) 415-424. ⇒155

[22] B.H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, et al., The multimodal brain tumor image segmentation benchmark (BRATS), *IEEE Transactions on Medical Imaging* **34**, 10 (2015) 1993-2024. ⇒164

[23] H.J. Park, S.M. Lee, J.W. Song, S.M. Lee, S.Y. Oh, N. Kim, J.B. Seo, Texture-Based Automated Quantitative Assessment of Regional Patterns on Initial CT in Patients With Idiopathic Pulmonary Fibrosis: Relationship to Decline in Forced Vital Capacity, *American Journal of Roentgenology* **207**, 5 (2016) 976-983. ⇒156

[24] S.C. Park, J. Tan, X.W. Wang, D. Lederman, J.K. Leader, S.H. Kim, B. Zheng, Computer-aided detection of early interstitial lung diseases using low-dose CT images, *Physics in Medicine & Biology* **56** (2011) 1139-1153. ⇒154

[25] C.A. Powell, S. Modi, H. Iwala, S. Takahashi, E.F. Smit, S. Siena, D.Y. Chang, E. Macpherson, A. Qin, J. Singh, C. Taitt, N. Shire, D. Ross Camidge, Pooled analysis of drug-related interstitial lung disease and/or pneumonitis in nine trastuzumab deruxtecan monotherapy studies, *ESMO Open* **7**, 4 (2022) 100544. ⇒148

[26] G. Raghu, M. Remy-Jardin, J.L. Myers, L. Richeldi, C.J. Ryerson, D.J. Lederer, et al., Diagnosis of idiopathic pulmonary fibrosis an official ATS/ERS/JRS/ALAT clinical practice guideline, *American Journal of Respiratory and Critical Care Medicine* **198**, 5 (2018) e44-e68. ⇒147, 150

[27] H.C. Shin, H.R. Roth, M.C Gao, L. Lu, Z.T. Xu, I. Nogues, J.H. Yao, D. Mollura, R.M. Summers, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, *IEEE Transactions on Medical Imaging* **35**, 5 (2016) 1285-1298. ⇒158, 160

[28] Y. Song, W.D. Cai, Y. Zhou, D.D. Feng, Feature-based image patch approximation for lung tissue classification, *IEEE Transactions on Medical Imaging* **32**, 4 (2013) 797-808. ⇒157

[29] C. Szegedy, W.Liu, Y.Q. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015) pp. 1-9. ⇒154

[30] A.A. Trușculescu, D. Manolescu, E. Tudorache, C. Oancea, Deep learning in interstitial lung disease – how long until daily practise, *European Radiology* **30**, (2020) 6285-6292. ⇒150

[31] A.A. Trușculescu, D.L. Manolescu, L. Broască, V.M. Ancușa, H. Ciocârlie, C.C. Pescaru, E. Vaștag, C.I. Oancea, Enhancing Imagistic Interstitial Lung Disease Diagnosis by Using Complex Networks, *Medicina (Kaunas)* **58**, 9 (2022) 1288. ⇒153, 154

[32] A. Wallis, K. Spinks, The diagnosis and management of interstitial lung diseases, *BMJ* **2015** (2015) 350:h2072. ⇒149

Received: June 29, 2023 • Revised: July 20, 2023

 Acta Univ. Sapientiae Informatica **15,** 1 (2023) 170–180

DOI: 10.2478/ausi-2023-0012

On connectivity of the semi-splitting block graph of a graph

Nivedha BASKAR
CHRIST(Deemed to be University)
Bengaluru, India
email: nivedha.b@res.christuniversity.in

Tabitha Agnes MANGAM
CHRIST(Deemed to be University)
Bengaluru, India
email:
tabitha.rajashekar@christuniversity.in

Mukti ACHARYA
CHRIST(Deemed to be University)
Bengaluru, India
email: mukti1948@gmail.com

Abstract. A graph G is said to be a semi-splitting block graph if there exists a graph H such that $S_B(H) \cong G$. This paper establishes a characterisation of semi-splitting block graphs based on the partition of the vertex set of G. The vertex (edge) connectivity and p-connectedness (p-edge connectedness) of $S_B(G)$ are examined. For all integers a, b with $1 < a < b$, the existence of the graph G for which $\kappa(G) = a$, $\kappa(S_B(G)) = b$ and $\lambda(G) = a, \lambda(S_B(G)) = b$ are proved independently. The characterization of graphs with $\kappa(S_B(G)) = \kappa(G)$ and a necessary condition for graphs with $\kappa(S_B(G)) = \lambda(S_B(G))$ are achieved.

1 Introduction

Graph theory has a wide range of applications in communication networks. An interconnection network can be represented as a simple connected graph

Key words and phrases: semi-splitting block graph of a graph, vertex connectivity, edge connectivity

$G = (V, E)$, where V represents the set of memory modules and E represents the communication links. The vertex cut of a graph G is the set of vertices whose removal gives a disconnected or trivial graph. The minimum cardinality of the vertex cut of the graph is called the vertex connectivity of the graph G, denoted by $\kappa(G)$. If $G - v$ has more than one component then v is a cut vertex of G. A maximal connected subgraph of the graph G which has no cut vertex is called a block of G. The edge cut of a graph G is the set of edges whose removal gives a disconnected graph. The minimum cardinality of the edge cut of the graph is called the edge connectivity of the graph G, denoted by $\lambda(G)$. A graph G is p-connected (p-edge connected) if $\kappa(G) \geq p$ ($\lambda(G) \geq p$).

Many results have been established regarding the connectivity of simple graphs, derived graphs and digraphs over many decades. This paper focuses on analyzing a derived graph's vertex(edge) connectedness, defined in [4], which is stated as follows.

Definition 1 *The semi-splitting block graph* $S_B(G)$ *of a graph of order* n *is a graph with* $V(S_B(G)) = V(G) \cup V_1(G) \cup B(G)$*, where*
$V(G) = \{v_i \mid 1 \leq i \leq n\}$,
$V_1(G) = \{u_i \mid 1 \leq i \leq n, v_i \in V(G)\}$,
$B(G) = \{b_l \mid 1 \leq l \leq k,\ B_l \text{ is a block in } G\}$.

$$E(S_B(G)) = \begin{cases} v_i v_j \mid 1 \leq i, j \leq n, v_i v_j \in E(G) \\ u_i v_j \mid 1 \leq i, j \leq n, v_i v_j \in E(G) \\ v_i b_l \mid 1 \leq i \leq n, 1 \leq l \leq k, v_i \in B_l \text{ in } G \end{cases}$$

where $v_i, v_j \in V(G)$, $u_i \in V_1(G)$ *and* $b_l \in B(G)$.

Since every edge is a block in a tree, the graph P_3 has 2 blocks say, $B_1 = \{v_1, v_2\}$ and $B_2 = \{v_2, v_3\}$. Figure 1, shows the semi-splitting block graph of P_3.

The study on the planarity of $S_B(G)$ has been carried out extensively in [4]. The scope of this paper is limited to simple, finite and undirected graphs. For terminology in graph theory, refer to [1, 2, 3].

2 Structural properties of $S_B(G)$

In this section, the structural properties of semi-splitting block graph of a graph are examined. If G is a disconnected graph with non trivial components $G_1, G_2, \ldots, G_m$, then $S_B(G)$ has $S_B(G_1), S_B(G_2), \ldots, S_B(G_m)$ as its components.

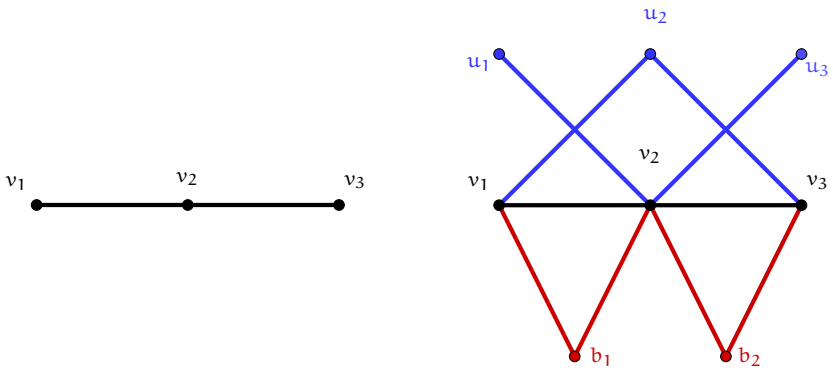


Figure 1: P_3 and $S_B(P_3)$

Theorem 2 *Let* $S_B(G)$ *be the semi-splitting block graph of* G *of order* n, $n \geq 2$, *and* k *blocks. Then,*

(i) For each $u_i \in V_1(G)$, $\deg_{S_B(G)}(u_i) = \deg_G(v_i)$, $v_i \in V(G)$, $1 \leq i \leq n$,

(ii) For each $b_l \in B(G)$, $\deg_{S_B(G)}(b_l) = |V(B_l)|$, *where* $1 \leq l \leq k$ *and* $V(B_l) \subset V(G)$,

(iii) For each $v_i \in V(G)$, $\deg_{S_B(G)}(v_i) = 2\deg_G(v_i) + s$, *where* s *is the number of blocks containing* v_i *in* G *and* $1 \leq i \leq n$.

Proof. Clearly, $N_{S_B(G)}(v_i) = N_G(v_i) \cup \{u_j : v_j \in N_G(v_i)\} \cup \{b_l : v_i \in B_l \text{ in } G\}$. $N_{S_B(G)}(u_i) = \{v_j : v_j \in N_G(v_i)\}$ and $N_{S_B(G)}(b_l) = V(B_l)$, where $V(B_l) \subset V(G)$. Hence the theorem follows. □

Corollary 3 *For a non-cut vertex* v_m *of the graph* G, $\deg_{S_B(G)}(v_m) = 2\deg_G(v_m) + 1$.

Corollary 4 *If* G *is a block of order* n, $n \geq 2$, *then*

(i) For each $v_i \in V(G)$, $\deg_{S_B(G)}(v_i) = 2\deg_G(v_i) + 1$, $1 \leq i \leq n$,

(ii) For $b_1 \in B(G)$, $\deg_{S_B(G)}(b_1) = n$.

Remark 5 $S_B(G)$ *is always non-regular, for any graph* G, *for all* $v_i \in V(G)$, $\deg_{S_B(G)}(v_i) > \deg_{S_B(G)}(u_i)$, $1 \leq i \leq n$.

The characterization of semi-splitting block graph based on the partition of the vertex set of the graph is given in the following theorem.

Theorem 6 *The following statements are equivalent.*

1. *A graph* G *of order* n *is a semi-splitting block graph.*
2. *The vertex set of a graph* G *can be partitioned into three subsets namely* V_1, V_2, V_3 *such that*
   - *(a) i. There is a bijective mapping* $f : V_1 \to V_2$ *such that* $f(v_1) = v_2$, *where* $v_1 \in V_1, v_2 \in V_2$.
     ii. $N(v_2) = N(v_1) \cap V_1$
   - *(b) For each* $v_3 \in V_3$, $< N(v_3) >$ *is a block of* $< V_1 >$.

Proof. $(1) \implies (2)$. Let G be a semi-splitting block graph of order n. Then for some H, $G \cong S_B(H)$. By definition, adjacency between two vertices in $S_B(H)$ is as follows:

I. adjacent vertices in H are adjacent in $S_B(H)$.

II. for each vertex v_i of $V(H)$, a new vertex u_i being adjacent to $N_H(v)$ is added.

III. for each block in H, a new vertex b_l adjacent to all the vertices of the respective block is added.

Let $V_1 = V(H)$, $V_2 = \{u_i\}_{i=1}^{|V(H)|}$ and $V_3 = \{b_l\}_{l=1}^{k}$. For each $v_i \in V_1$, let $u_i \in V_2$ be the corresponding new vertex added in $S_B(H)$. Then, $f : V_1 \to V_2$ is a bijective mapping such that $f(v_i) = u_i$ and $N(u_i) = N(v_i) \cap V_1$. Since, each $b_l \in V_3$ is adjacent only to vertices of the corresponding block, $N(b_l) = B_l$, where $1 \le l \le k$ and B_l is a unique block of $< V_1 >$.
$(2) \implies (1)$. Suppose (2) is true. Let $H =< V_1 >$. Then, $G \cong S_B(H)$.
Therefore, G is a semi-splitting block graph.
Hence the theorem. □

Theorem 7 *For any graph* G *of order* n, $n \ge 2$, *with* k *blocks,* $\delta(S_B(G)) = \min\{|V(B_l)|, \delta(G)\}$, *where* B_l, $1 \le l \le k$ *is a block of* G.

Proof. It is evident from Theorem 2 that in $S_B(G)$, for any $v_i \in V(G)$, $\deg(u_i) = \deg_G(v_i)$ and $\deg(v_i) > 2\deg_G(v_i)$. The following cases are considered.

Case 1 Suppose G is a block. By Corollary 4, $\deg(b_1) = n > \delta(G)$ in $S_B(G)$. Therefore, $\delta(S_B(G)) = \delta(G)$.

Case 2 Suppose G is not a block. Then there exist at least two blocks in G. This implies that $|V(B_l)| \geq 2$, for $1 \leq l \leq k$ and $k \geq 2$.
Thus, $\delta(S_B(G)) = \min\{\delta(G), |V(B_l)|\}$.

Therefore, it inferred that, $\delta(S_B(G)) = \min\{\delta(G), |V(B_l)|\}$.
Hence the theorem. □

Corollary 8 *For any graph* G, $\delta(S_B(G)) = \delta(G)$ *if and only if* G *is a block or* G *is not a block with* $|V(B_l)| \geq \delta(G)$ *for all* $1 \leq l \leq k$.

Corollary 9 *For integers* a, b *with* $a > b > 1$, *there exists a graph* G *with* $\delta(G) = a$ *and* $\delta(S_B(G)) = b$ *if and only if there exists at least one block* B_l *in* G *whose* $|V(B_l)| < \delta(G)$, *where* $1 \leq l \leq k$.

3 Connectedness of $S_B(G)$

In this section, the vertex (edge) connectedness of semi-splitting block graph of a connected graph is examined. Let $v_a \in V(G)$ such that $\deg(v_a) = \delta(G)$. Since, $\kappa(G) \leq \lambda(G) \leq \delta(G)$, $N(v_a) = \{v_s | 1 \leq s \leq \delta(G)\}$ is a vertex cut of G and $Y = \{(v_a, v_s) | v_s \in N(v_a)\}$ is an edge cut in G.

Theorem 10 *If* G *is a block with* $\deg(v_a) = \delta(G)$, *then the following statements are true in* $S_B(G)$.

1. $N(u_a)$ *is a vertex cut.*
2. $Y' = \{(u_a, v_s) | v_s \in N_G(v_a)\}$ *is an edge cut.*

Proof. Consider G to be a block such that $\deg(v_a) = \delta(G)$. By Theorem 2 and Corollary 8, $\deg(u_a) = \delta(S_B(G))$. As $\kappa(S_B(G)) \leq \lambda(S_B(G) \leq \delta(S_B(G))$ and $N(u_a) = N_G(v_a)$, $N(u_a)$ is a vertex cut and $Y' = \{(u_a, v_s) | v_s \in N(u_a)\}$ is an edge cut in $S_B(G)$.
Hence the theorem. □

Let $S = \{v_j | 1 \leq j \leq t\}$ be a minimum vertex cut of G. As G is an induced subgraph of $S_B(G)$, S is the subset of a vertex cut of $S_B(G)$. The following theorem gives the vertex connectivity of $S_B(G)$.

Theorem 11 *For a connected graph* G *with order* n, $n \geq 2$,

$$\kappa(S_B(G)) = \begin{cases} \min\{\delta(G), 2\kappa(G) + 1\} & \text{when } G \text{ is a block} \\ \min\{2, \delta(G)\} & \text{otherwise} \end{cases}$$

Proof. Let G be a connected graph of order $n \geq 2$. The following cases are considered:

Case 1 Suppose G is not a block.
Let v_c be a cut vertex in G. In $S_B(G)$, the vertices of $N_G(v_c)$ are adjacent to v_c and u_c. Thus, u_c is the cut vertex in $S_B(G)$ if and only if v_c is a pendant vertex in G. In all other cases, removing the vertices $\{v_c, u_c\}$ disconnects the graph $S_B(G)$. Thus, $\kappa(S_B(G)) = 1$, if $\delta(G) = 1$ and $\kappa(S_B(G)) = 2$, otherwise.
Hence, $\kappa(S_B(G)) = \min\{2, \delta(G)\}$.

Case 2 Suppose G is a block.
In $S_B(G)$, there exists exactly one block vertex b_1 adjacent to all the vertices of G. Thus, $S' = S \cup \{u_j, b_1 | 1 \leq j \leq t\}$ and by Theorem 10, $N(u_a)$ are vertex cuts in $S_B(G)$. Here, $|S'| = 2\kappa(G) + 1$ and $|N(u_a)| = \delta(S_B(G)) = \delta(G)$.
Hence, $\kappa(S_B(G)) = \min\{\delta(G), 2\kappa(G) + 1\}$.

Hence the theorem. □

Corollary 12 *For a connected graph* G, $\kappa(S_B(G)) = \kappa(G)$ *if and only if* $\kappa(G) = \delta(G)$.

Proof. Suppose $\kappa(G) = \delta(G)$.
Suppose G is a block. As $\delta(G) < 2\delta(G) + 1$, by Theorem 11, $\kappa(S_B(G)) = \delta(G) = \kappa(G)$. If G is not a block, then $\kappa(G) = 1$. By Theorem 11, $\kappa(S_B(G)) = \delta(G) = \kappa(G)$. The converse can also be proved in the same manner.
Hence the theorem. □

The following theorem gives the necessary and sufficient condition for the existence of a graph whose vertex connectivity is a and the vertex connectivity of its $S_B(G)$ is b, for all a, b such that $1 < a < b$.

Theorem 13 *For integers* a, b *with* $1 < a < b$, *there exists a graph* G *with* $\kappa(G) = a$ *and* $\kappa(S_B(G)) = b$ *if and only if* $b \leq 2a + 1$.

Proof. Assume that $b \leq 2a + 1$. Consider G_1 and G_2 as any two connected block graphs each of minimum degree b. The following assumptions are made.

1. $V(G_1) = \{v_{1_r} | 1 \leq r \leq s, s > b\}$ and
   $V(G_2) = \{v_{2_w} | 1 \leq w \leq t, t > b\}$.
2. Let $\deg(v_{x_y}) = b$, where $v_{x_y} \in V(G_1 \cup G_2)$.

The graph G is formed from G_1 and G_2 by adding (v_{1_q}, v_{2_q}) new edges such that $v_{1_q}, v_{2_q} \neq v_{x_y}$ and $1 \leq q \leq a$. Here, G is a block with $deg(v_{x_y}) = b$. The removal of vertices v_{1_q}, $1 \leq q \leq a$, disconnects the graph G. Thus, $X = \{v_{1_q} | 1 \leq q \leq a\}$ and $N_G(v_{x_y})$ are the vertex cuts of G which implies, $\kappa(G) = min\{|X|, |N_G(v_{x_y})|\}$. As $|X| = a$ and $|N_G(v_{x_y})| = b > a$, we get $\kappa(G) = |X| = a$. Hence, $X' = X \cup \{u_{1_q}, b_1 | 1 \leq q \leq a\}$ and by Theorem 10, $N(u_{x_y})$ are the vertex cuts in $S_B(G)$, where $deg(u_{x_y}) = \delta(S_B(G))$. Therefore, $\kappa(S_B(G)) = min\{|X'|, |N(u_{x_y})|\}$. Since, $|X'| = 2a+1$ and $|N(u_{x_y})| = b \leq 2a+1$, we conclude that $\kappa(S_B(G)) = b$.

On the contrary, consider $b > 2a + 1$. Let G be a graph as defined above, then $\kappa(G) = a$ and $\kappa(S_B(G)) = min\{|X'|, |N(u_{x_y})|\}$. Here, $|X'| = 2a + 1$ and $|N(u_{x_y})| = b > 2a + 1$, which implies that $\kappa(S_B(G)) = 2a + 1$.

Therefore, $\kappa(S_B(G)) \neq b$.

Hence the theorem. □

Let $\lambda(G) = t$, i.e., $T = \{(v_e, v_f) | (v_e, v_f) \in E(G), e \neq f\}$ be a minimum edge cut of G. Since G is an induced subgraph of $S_B(G)$, T is the subset of an edge cut of $S_B(G)$. The edge connectivity of $S_B(G)$ is discussed in the next theorem.

Theorem 14 *For a connected graph of order* n, $n \geq 2$,

1. *If* G *has a bridge, then* $\lambda(S_B(G)) = min\{2, \delta(G)\}$.
2. *If* G *is a block, then* $\lambda(S_B(G)) = \delta(G)$.

Proof. Let G be a connected graph with $n \geq 2$. The following cases are considered:

Case 1 Suppose G has a bridge.

Let $e_m = \{v_g, v_h\}$ be a bridge. As every bridge is a block, let B_m be a block with $V(B_m) = \{v_g, v_h\}$. Thus, $U = \{(b_m, v_g), (b_m, v_h)\}$ is an edge cut in $S_B(G)$, where $b_m \in V(S_B(G)))$ and $|U| = 2$. By Theorem 7, e_m is a bridge in $S_B(G)$ if and only if $\delta(G) = 1$, which implies that $\lambda(S_B(G)) = 1$, when $\delta(G) = 1$. Suppose $\delta(G) \geq 2$, then $\delta(S_B(G)) \geq 2$. Hence, U is the minimum edge cut in $S_B(G)$, for $\delta(G) \geq 2$. Therefore, it is concluded that $\lambda(S_B(G)) = min\{2, \delta(G)\}$.

Case 2 Suppose G is a block.

Then, $\lambda(G) \geq 2$. In $S_B(G)$, b_1 is the only block vertex such that $|N(b_1)| = n$. Thus, $T' = T \cup \{(u_e, v_f), (v_e, u_f), (b_1, v_i) | 1 \leq i \leq n\}$ and by Theorem 10, $Y' = \{(u_a, v_k) | v_k \in N_G(v_a)\}$ are edge cuts in $S_B(G)$. Hence, $\lambda(S_B(G)) = \{|T'|, |Y'|\}$. Here, $|T'| = 3\lambda(G) + n$ and $|Y'| = \delta(G) < n$. Therefore, $\lambda(S_B(G)) = \delta(G)$.

Hence the theorem. □

Note that from Theorem 14 (2), $\lambda(G) = a \geq 2$ and $\lambda(S_B(G)) = \delta(G) \geq a$. This leads to the following corollary.

Corollary 15 *For integers* a, b *with* $1 < a \leq b$*, there exists a graph* G *with* $\lambda(G) = a$ *and* $\lambda(S_B(G)) = b$.

Theorem 16 *Let* G *be a connected graph which is bridgeless and not a block. If* G *has* $T = \{(v_c, v_f) | 1 \leq f \leq t, v_c \text{ is a cut vertex}\}$ *as a minimum edge cut, then* $\lambda(S_B(G)) = \min\{\delta(S_B(G)), 3\lambda(G) + 1\}$.

Proof. Consider G to be a connected graph which is bridgeless and not a block. Let T be a minimum edge cut. Thus in $S_B(G)$, $T' = T \cup \{(v_c, u_f), (u_c, v_f), (v_c, b_x)\}$, where $v_c, v_f \in B_x$ and by Theorem 10, $Y' = \{(v_a, v_s) | v_s \in N(v_a), \deg(v_a) = \delta(G)\}$ are edge cuts. Here, $|T'| = 3\lambda(G) + 1$.
Therefore, $\lambda(S_B(G)) = \min\{\delta(S_B(G)), 3\lambda(G) + 1\}$. □

For any block $B_l, 1 \leq l \leq k$, in G, $|V(B_l)| \geq 2$ if G has a bridge and $|V(B_l)| \geq 3$, otherwise. The following theorem gives a necessary condition for $S_B(G)$ for which its vertex and edge connectivity are equal.

Theorem 17 *If* $\Delta(G) \leq 3$*, then* $\kappa(S_B(G)) = \lambda(S_B(G))$.

Proof. Let G be a connected graph of order $n \geq 2$ and $\Delta(G) \leq 3$. The following cases are considered:

Case 1 Suppose $\delta(G) \leq 2$.
By Theorem 11, $\kappa(S_B(G))) = \delta(G)$. When G is bridgeless and is not a block, by Theorem 14, $\lambda(S_B(G)) = \delta(G)$. Consider a graph G which is bridgeless and has a cut vertex then, $\lambda(G) \geq 2$. Since $\lambda(G) \leq \delta(G)$, $\lambda(G) = \delta(G) = 2$. As $|V(B_l)| \geq 3$, for all $1 \leq l \leq k$, by Theorem 7, $\delta(S_B(G)) = \delta(G)$ in $S_B(G)$. Since $\lambda(S_B(G)) \leq \delta(S_B(G))$, it implies that $\lambda(S_B(G)) \leq 2$. The graph G being an induced subgraph of $S_B(G)$, $\lambda(G) \leq \lambda(S_B(G))$ and $\lambda(S_B(G)) = 2 = \delta(G)$.
Therefore, $\kappa(S_B(G)) = \lambda(S_B(G))$.

Case 2 Since $\Delta(G) = 3$, G is 3-regular. If G is a block, then by Theorems 11 and 14, $\kappa(S_B(G)) = \lambda(S_B(G)) = \delta(G)$. When G is 3-regular, G has a bridge if and only if G has a cut vertex. So, if G is not a block, then G has a bridge. Thus, by Theorems 11 and 14, $\kappa(S_B(G)) = \lambda(S_B(G)) = 2$.

Therefore, $\kappa(S_B(G)) = \lambda(S_B(G))$.
Hence the theorem. □

A graph G is p-connected (p-edge connected) if and only if every pair of vertices is joined by at least p vertex (edge) disjoint paths. Also, every p-connected graph is p-edge connected. Further, G is p-edge connected if and only if each of its blocks is p-edge connected. It follows that G is p-connected if and only if each of its blocks is p-edge connected. So, $\lambda(B_l) \geq k$, which implies $\delta(B_l) \geq k$, where $1 \leq l \leq k$. Now, the p-connectedness (p-edge connected) of $S_B(G)$ is discussed.

Theorem 18 *If G is p-connected (p-edge connected) with $p \geq 1$, then $S_B(G)$ is also p-connected (p-edge connected).*

Proof. Let G be a p-connected (p-edge connected) graph with $p \geq 1$. To prove $S_B(G)$ is p-connected (p-edge connected) it is enough to show that between any two vertices of $S_B(G)$, there exist p-vertex (edge) disjoint paths. The following cases are considered:

Case 1 Let $v_a, v_b \in V(G)$
Then, v_a and v_b have at least p vertex (edge) disjoint paths between them. Since G is an induced subgraph of $S_B(G)$, in $S_B(G)$ there exist at least p disjoint paths between the vertices v_a and v_b.

Case 2 Let $v_a \in V(G)$ and $u_b \in V_1(G)$
Since $p \leq \kappa(G) \leq \delta(G)$, there exist at least p vertices adjacent to v_a. Assume that $v_{a_1}, v_{a_2}, \ldots, v_{a_p}$ are the vertices adjacent to v_a in G. Then in $S_B(G)$, by Case 1 that there exist p vertex (edge) disjoint paths between v_a and v_{a_i} ($1 \leq i \leq p$). In addition, the vertices $u_{a_i} \in V_1(G)$, $1 \leq i \leq p$, are adjacent to the vertex v_a. As $u_b \in V_1(G)$, by Theorem 6, $N(u_b) = N(v_b) \cap V(G)$ for some $v_b \in V(G)$. Since $|N(v_b)| \geq p$, let $v_{b_1}, \ldots, v_{b_p} \in V(G)$ such that $v_{b_i} \in N(v_b)$ for $1 \leq i \leq p$, which implies in $S_B(G)$, $v_{b_i} \in N(u_b)$. As $v_a, v_{b_i} \in V(G)$, by Case 1 in $S_B(G)$ there exist p vertex (edge) disjoint paths between them. In conclusion, the vertices v_a, u_b are joined by p vertex (edge) disjoint paths between them in $S_B(G)$.

Case 3 Let $v_a \in V(G)$, $b_m \in B(G)$.
As defined in Case 2, let $v_{a_i} \in N(v_a)$, $1 \leq i \leq p$. Since $\delta(B_m) \geq k$, we have $|V(B_m)| > k$. Let $v_{x_1}, \ldots, v_{x_p} \in V(G)$ such that $v_{x_i} \in N(b_m)$, $1 \leq i \leq p$. It follows from Case 1 that in $S_B(G)$, the vertices v_a and v_{x_i} have p vertex (edge) disjoint paths between them as $v_a, v_{x_i} \in V(G)$. Therefore, in $S_B(G)$, the vertices v_a, b_m are joined by p vertex (edge) disjoint paths between them.

Case 4 Let $u_a, u_b \in V_1(G)$
As defined in Case 2, let $v_{a_i} \in N(u_a)$ and $v_{b_i} \in N(u_b)$, $1 \leq i \leq p$. Since $v_{a_i}, v_{b_i} \in V(G)$ and by Case 1, the theorem follows.

Case 5 Let $u_a \in V_1(G)$, $b_m \in B(G)$
It follows from Case 2 and 3, let $v_{a_i} \in N(u_a)$ and $v_{x_i} \in N(b_m)$, $1 \leq i \leq p$, respectively. As $v_{a_i}, v_{x_i} \in V(G)$ and by Case 1, the theorem follows.

Case 6 Let $b_m, b_s \in B(G)$
It follows from Case 3, let $v_{x_i} \in N(b_m)$ and $v_{y_i} \in N(b_s)$, $1 \leq i \leq p$. As $v_{x_i}, v_{y_i} \in V(G)$ and by Case 1, the theorem follows.

Hence the theorem. □

If G is p-connected, then $G + K_1$ is $(p+1)$-connected. This result leads to the following theorem.

Theorem 19 *If G is p-connected, then the following statements are true:*

1. *$S_B(G + K_1)$ is $(p+1)$-connected.*
2. *$S_B(G) + K_1$ is $(p+1)$-connected.*

Proof. Consider the graph G to be a p-connected. Then, $G + K_1$ is $(p+1)$-connected. Also, by Theorem 18, $S_B(G)$ is p-connected.
Hence the theorem follows. □

Conclusion

The structural properties of $S_B(G)$ have been investigated which helped to determine the results on the vertex and edge connectivity of $S_B(G)$. The semi-splitting block graph has been characterized based on its vertex set. A necessary condition for p-connectedness (p-edge connectedness) of $S_B(G)$ has been established. The scope of future work is to characterize graphs whose $S_B(G)$ is $(p+1)$-connected ($(p+1)$-edge connected).

References

[1] F. Buckley, F. Harary, *Distance in graphs*, Addison–Wesley, Redwood City, 1990. ⇒171

[2] F. Harary, *Graph Theory*, Addison–Wesley, Reading, MA, 1990. ⇒171

[3] D. B. West, *Introduction to Graph Theory* (2nd ed.), Pearson Education, 2002. ⇒171

[4] V. R. Kulli, K. M. Niranjan The semi-splitting block graph of a graph, *Journal of Scientific Research,* **2,** 3 (2010) 485–488. ⇒171

Received: May 30, 2023 • *Revised: July 21, 2023*