Computer Science Journal of Moldova, vol.**27**, no.1(79), 2019

Graphs with Large Hop Roman Domination Number

E. Shabani, N. Jafari Rad, A. Poureidi

Abstract

A subset S of vertices of a graph G is a hop dominating set if every vertex outside S is at distance two from a vertex of S. A Roman dominating function on a graph $G = (V, E)$ is a function $f : V(G) \longrightarrow \{0, 1, 2\}$ satisfying the condition that every vertex u for which $f(u) = 0$ is adjacent to at least one vertex v for which $f(v) = 2$. A hop Roman dominating function (HRDF) of G is a function $f : V(G) \longrightarrow \{0, 1, 2\}$ having the property that for every vertex $v \in V$ with $f(v) = 0$ there is a vertex u with $f(u) = 2$ and $d(u, v) = 2$. The weight of a HRDF f is the sum $f(V) = \sum_{v \in V} f(v)$. The minimum weight of a HRDF on G is called the hop Roman domination number of G and is denoted by $\gamma_{hR}(G)$. In this paper we characterize all graphs G of order n with $\gamma_{hR}(G) = n$ or $\gamma_{hR}(G) = n - 1$.

Keywords: Domination, Roman domination, Hop Roman domination.

MSC 2010: 05C69.

1 Introduction

For notation and graph theory terminology not given here, we refer to [7]. Let $G = (V, E)$ be a graph with vertex set $V = V(G)$ and edge set $E = E(G)$. The *order* of G is $n(G) = |V(G)|$. The *open neighborhood* of a vertex v is $N_G(v) = \{u \in V(G) \,|\, uv \in E(G)\}$. The *degree* of v, denoted by $\deg(v)$, is $|N_G(v)|$. The *open neighborhood* of a subset $S \subseteq V$, is $N_G(S) = \bigcup_{v \in S} N_G(v)$, and the *closed neighborhood* of S is the set $N_G[S] = N_G(S) \cup S$. The *distance* between two vertices u

and v in G, denoted by $d(u,v)$, is the minimum length of a (u,v)-path in G. The diameter, $diam(G)$, of G is the maximum distance among all pairs of vertices in G. If S is a subset of vertices in a graph G, then we denote by $G[S]$ the subgraph of G induced by S. For an integer $k \geq 1$, the set of all vertices at distance k from v is denoted by $N_k(v)$. Also we denote $N_k[v] = N_k(v) \cup \{v\}$.

A subset of vertices of a graph G is a dominating set of G if every vertex in $V(G)-S$ has a neighbor in S. The domination number, $\gamma(G)$, is the minimum cardinality of a dominating set of G. Ayyaswamy and Natarajan [4] introduced the concept of hop domination in graphs. A subset S of vertices of a graph G is a *hop dominating set* (HDS) if every vertex outside S is at distance two from a vertex of S. The *hop domination number*, $\gamma_h(G)$, of G is the minimum cardinality of a hop dominating set of G. A HDS of G of minimum cardinality is referred as a $\gamma_h(G)$-set. Farhadi et al. [6] generalized hope dominating sets and studied k-hop dominating sets for every integer $k \geq 2$. The concept of hop domination was further studied, for example, in [3], [8], [9]. We define the *hop-degree* of a vertex v in a graph G, denoted $\deg_h(v)$, to be the number of vertices at distance 2 from v in G. The maximum hop-degree among the vertices of G is denoted by $\Delta_h(G)$.

A function $f : V(G) \longrightarrow \{0,1,2\}$ having the property that for every vertex $v \in V$ with $f(v) = 0$, there exists a vertex $u \in N(v)$ with $f(u) = 2$, is called a *Roman dominating function* or just an RDF. The *weight* of an RDF f is the sum $f(V) = \sum_{v \in V} f(v)$. The minimum weight of an RDF on G is called the *Roman domination number* of G and is denoted by $\gamma_R(G)$. The mathematical concept of Roman domination was defined and discussed by Stewart [13], and ReVelle and Rosing [11], and subsequently developed by Cockayne et al. [5].

Roman dominating functions with further properties were considered by several authors, see for example, [1], [2], [10]. For any RDF f on a graph G, it is clear that $\{v \in V(G) : f(v) \neq 0\}$ is a dominating set for G. It is an interesting question to study those RDF f such that the set $\{v \in V(G) : f(v) \neq 0\}$ is a hop dominating set for G. Shabani [12] considered Roman dominating functions with the above property and introduced the concept of hop Roman dominating func-

tions. A *hop Roman dominating function* (HRDF) on a graph G is a function $f : V(G) \longrightarrow \{0, 1, 2\}$ having the property that for every vertex $v \in V$ with $f(v) = 0$ there is a vertex u with $f(u) = 2$ and $d(u, v) = 2$. The weight of a HRDF f is the sum $f(V) = \sum_{v \in V} f(v)$. The minimum weight of a HRDF on G is called the *Hop Roman domination number* of G and is denoted $\gamma_{hR}(G)$. A HRDF with minimum weight is referred as a $\gamma_{hR}(G)$-function. For a HRDF f in a graph G, we denote by V_i (or V_i^f to refer to f) the set of all vertices of G with label i under f. Thus a HRDF f can be represented by a triple (V_0, V_1, V_2), and we can use the notation $f = (V_0, V_1, V_2)$. Among other results, Shabani et al. [12] obtained the following.

Proposition 1 (Shabani [12]) *For any graph G of order n and with $\Delta_h(G) \geq 1$, $\gamma_{hR}(G) \geq \dfrac{2n}{\Delta_h(G) + 1}$.*

Proposition 2 (Shabani [12]) *For any graph G of order n and with $\Delta_h(G) \geq 1$, $\gamma_{hR}(G) \leq n - \Delta_h(G) + 1$.*

In this paper, we characterize graphs with large hop Roman domination number. In Section 2, we characterize all graphs G of order n with $\gamma_{hR}(G) = n$. In Section 3, we characterize all graphs G of order n with $\gamma_{hR}(G) = n - 1$.

2 Graphs G with $\gamma_{hR}(G) = n$

In this section we characterize all graphs G of order n with $\gamma_{hR}(G) = n$. For this purpose, we define a family of graphs as follows. Let $\mathcal{G}$ be the family of graphs G such that G can be obtained from a sequence $G_0, G_1, ..., G_k$, $(k \geq 1)$, of graphs, where G_0 is a complete graph or $G_0 = \overline{K_2}$, $G = G_k$ and, if $k \geq 1$, then G_{i+1} can be obtained recursively from G_i by the following operation, for $i = 0, 1, ..., k - 1$.

Operation $\mathcal{O}$. Add two new vertices and join each new vertex to every vertex of G_i.

Theorem 3 *Let G be a connected graph of order n. Then $\gamma_{hR}(G) = n$ if and only if $G \in \mathcal{G} \cup \{P_4, K_n\}$.*

Proof. ($\Rightarrow$) The proof is by an induction on the order n of a graph G with $\gamma_{hR}(G) = n$. If $n \leq 2$ then clearly G is a complete graph. Thus assume that $n \geq 3$. Assume that the result is true for all graphs of order less than n and let G be a graph of order n with $\gamma_{hR}(G) = n$. From Proposition 2, we obtain that $\Delta_h(G) \leq 1$. Thus $diam(G) \leq 3$. If $diam(G) = 1$, then G is a complete graph, as desired.

Assume that $diam(G) = 2$. Let $v_1v_2v_3$ be a diametrical path in G. Assume that there exists a vertex $v \in N(v_1)$ such that $v \notin N(v_3)$. Since $diam(G) = 2$, we have $v \in N_2(v_3)$. Then $\deg_h(v_3) \geq 2$, a contradiction. Thus $N(v_1) \subseteq N(v_3)$, and similarly $N(v_3) \subseteq N(v_1)$. Consequently, $N(v_1) = N(v_3)$. If there are three vertices $a, b, c \in N(v_1)$ such that $a \notin N(b) \cup N(c)$, then $\deg_h(a) \geq 2$, a contradiction. Thus any vertex of $N(v_1)$ is adjacent to at least $|N(v_1)| - 2$ vertices of $N(v_1)$, and so $\delta(G[N(v_1)]) \geq |N(v_1)| - 2$. If $\delta(G[N(v_1)]) = |N(v_1)| - 1$, then $G[N(v_1)]$ is a complete graph. Thus G is obtained from $G[N(v_1)]$ by using Operation $\mathcal{O}$, and so $G \in \mathcal{G}$. Therefore assume that $\delta(G[N(v_1)]) = |N(v_1)| - 2$. Let $G' = G[N(v_1)]$. If $|V(G')| = 2$, then $G' = \overline{K_2}$, and so G is obtained from $\overline{K_2}$ by using Operation $\mathcal{O}$ and so $G \in \mathcal{G}$. Now, let $|V(G')| \geq 3$. Suppose that $\gamma_{hR}(G') \leq |V(G')| - 1$. Let f be a $\gamma_{hR}(G')$-function. Then $g = (V_0^f, V_1^f \cup \{v_1, v_3\}, V_2^f)$ is a HRDF of G with $g(V) \leq |V(G')| + 1$. Therefore, $\gamma_{hR}(G) \leq n - 1$, a contradiction. Therefore, $\gamma_{hR}(G') = |V(G')|$. By the inductive hypothesis, $G' \in \mathcal{G}$ or $G' \in \{P_4, K_n\}$. Since $\delta(G') = |V(G')| - 2$ we have $G' \in \mathcal{G}$. Therefore, G is obtained from $G' \in \mathcal{G}$ by using Operation $\mathcal{O}$ and so $G \in \mathcal{G}$.

It remains to assume that $diam(G) = 3$. Let $P : v_1v_2v_3v_4$ be a diametrical path in G. Suppose that $G \neq P_4$. Let $v \in V(G) - V(P)$. Clearly, v is not adjacent to both v_1 and v_4. Without loss of generality, assume that $v_1 \notin N(v)$. Since $diam(G) = 3$ and G is connected, we have $N(v) \cap \{v_2, v_3\} \neq \emptyset$. If $v_2 \in N(v)$, then $\deg_h(v_1) \geq 2$, a contradiction. Thus, $v_3 \in N(v)$, and so $\deg_h(v_2) \geq 2$, a contradiction. Thus $V(G) - V(P) = \emptyset$. Consequently, $G = P_4$.

($\Leftarrow$) Let $G \in \mathcal{G} \cup \{P_4, K_n\}$. We show that $\gamma_{hR}(G) = n$. If $G = K_n$ or $G = P_4$, then we can easily see that $\gamma_{hR}(G) = n$. Now suppose that $G \in \mathcal{G}$. There is a sequence of graphs $G_0, G_1, ...G_k$, $(k \geq 1)$, where G_0 is a complete graph or $G_0 = \overline{K_2}$, $G = G_k$ and, if $k \geq 1$, then G_{i+1} can be

obtained recursively from G_i by using Operation $\mathcal{O}$ for $i = 0, ..., k-1$. We use an induction on k to show that $\delta(G) = |V(G)| - 2 = n - 2$. If $k = 1$, then $G = C_4$ or G is obtained from the complete graph K_{n-2} and we can easily see that $\delta(G) = n-2$. This establishes the basic step. Suppose now that $k \geq 2$ and the result is true for all graphs $G \in \mathcal{G}$ that can be constructed from a sequence of length at most $k-1$, and let $G^{'} = G_{k-1}$. By the induction hypothesis, $\delta(G^{'}) = |V(G^{'})| - 2$. By construction, G is obtained from $G^{'}$ by using Operation $\mathcal{O}$. Let x and y be two vertices joined to any vertex of $G^{'}$ according to the Operation $\mathcal{O}$. Then, clearly $d(x, y) = 2$ and $\deg(x) = \deg(y) = |V(G^{'})| = n - 2$. Now let $x^{'} \in V(G^{'})$ be a vertex with $\deg_{G^{'}}(x^{'}) = \delta(G^{'}) = |V(G^{'})| - 2$. Then $\deg_G(x^{'}) = \delta(G^{'}) + 2 = |V(G^{'})| = n - 2$. We conclude that $\delta(G) = n - 2$.

Now suppose, to the contrary, that $\gamma_{hR}(G) \leq n - 1$. From Proposition 2 we have $\Delta_h(G) \leq 2$. If $\Delta_h(G) = 2$, then there exists a vertex $w \in V(G)$ such that $\deg_h(w) = 2$ and so $\deg(w) \leq n - 3$, a contradiction with $\delta(G) = n - 2$. Thus $\Delta_h(G) = 1$. By Propositions 1 and 2 we have $\gamma_{hR}(G) = n$, a contradiction. We conclude $\gamma_{hR}(G) = n$. ■

3 Graphs G with $\gamma_{hR}(G) = n - 1$

In this section we characterize connected graphs G of order n with $\gamma_{hR}(G) = n - 1$. For this purpose, we define a family of graphs as follows. Let $\mathcal{G}$ be the family of graphs described in Section 2. Let $\mathcal{G}^*$ be the family of graphs G that can be obtained from a graph G', where G' is a complete graph or $G' \in \mathcal{G}$, by one of the following operations:

Operation $\mathcal{O}_1$: Add three vertices and join each of them to every vertex of G'.

Operation $\mathcal{O}_2$: Add a path $P_2 : v_1v_2$ and a vertex v, join v to every vertex of G' and join any vertex of $V(P_2)$ to at least $|V(G')| - 1$ vertices of G', such that $|V(G^{'})| \geq 2$ and if $G' \in \mathcal{G}$ and x and y are two vertices of G' with $d(x, y) = 2$, then the following conditions hold.

(i) $\{x, y\} \subseteq N(v_1) \cup N(v_2)$.

(ii) If $x \notin N(v_1) \cap N(v_2)$ or $y \notin N(v_1) \cap N(v_2)$, then v_1 and v_2 are adjacent to any vertex of $N_{G'}(x)$.

Lemma 4 *If $G \in \mathcal{G}^*$, then $\gamma_{hR}(G) = |V(G)| - 1$.*

Proof. Let $G \in \mathcal{G}^*$. We have the following cases.

Case 1. G is obtained from $G^{'}$ by using Operation $\mathcal{O}_1$. Let G be obtained from $G^{'}$ by joining three vertices x, y, z to any vertex of $G^{'}$. Clearly the distance between any pair of vertices of $\{x, y, z\}$ is two. Let $f^{'}$ be a $\gamma_{hR}(G^{'})-$function. By Theorem 3, $f^{'}(V) = |V(G^{'})|$. Then g defined by $g(x) = 2$, $g(y) = g(z) = 0$, and $g(u) = f^{'}(u)$ otherwise, is a HRDF for G of weight $f^{'}(V) + 2$. Thus $\gamma_{hR}(G) \leq f^{'}(V) + 2 = |V(G^{'})| + 2 = |V(G)| - 1$. On the other hand, let f be a $\gamma_{hR}(G)$-function. Then $f(x) + f(y) + f(z) \geq 2$ (otherwise, at least two vertices of $\{x, y, z\}$ are not hop Roman dominated by f, a contradiction). Then clearly $f|_{V(G')}$ is a HRDF for $G^{'}$. Then $\sum_{u \in V(G')} f(u) \geq \gamma_{hR}(G') = |V(G^{'})|$. Then $\gamma_{hR}(G) = w(f) \geq |V(G^{'})| + 2 = |V(G)| - 1$. Therefore, $\gamma_{hR}(G) = |V(G)| - 1$.

Case 2. G is obtained from $G^{'}$ by using Operation $\mathcal{O}_2$. Let v be the added vertex and v_1v_2 be the added P_2-path according to the Operation $\mathcal{O}_2$. Clearly the distance between any vertex of $V(P_2)$ and the vertex v is equal to two. Let $f^{'}$ be a $\gamma_{hR}(G^{'})-$function. By Theorem 3, $f^{'}(V) = |V(G^{'})|$. Then g defined by $g(v) = 2$, $g(v_1) = g(v_2) = 0$ and $g(z) = f^{'}(z)$ otherwise, is a HRDF for G with weight $f^{'}(V) + 2$. Thus $\gamma_{hR}(G) \leq f^{'}(V) + 2 = |V(G^{'})| + 2 = |V(G)| - 1$.

Claim 1. There is a $\gamma_{hR}(G)-$function, say as f, such that $f(v) + f(v_1) + f(v_2) \geq 2$ and also $\sum_{u \in V(G')} f(u) \geq |V(G^{'})|$.

Proof of Claim 1: Suppose that at least one vertex of $V(P_2)$ is adjacent to any vertex of $G^{'}$. Let g be a $\gamma_{hR}(G)-$function. Then $g(v) + g(v_1) + g(v_2) \geq 2$ (otherwise, at least one vertex of $\{v, v_1, v_2\}$ is not hop Roman dominated by g, a contradiction). If two vertices v_1 and v_2 are adjacent to any vertex of $G^{'}$, then clearly $g|_{V(G')}$ is a HRDF for $G^{'}$. Then $\sum_{u \in V(G')} g(u) \geq \gamma_{hR}(G') = |V(G^{'})|$. Thus, without loss of generality suppose that v_1 is adjacent to any vertex of $G^{'}$ and v_2 is adjacent to $|V(G^{'})| - 1$ vertices of $G^{'}$. Let a be a vertex of $G^{'}$ such that $a \notin N(v_2)$. Clearly $a \in N_2(v_2)$. If $g(a) \neq 0$, then clearly $g|_{V(G')}$ is a HRDF for $G^{'}$. Thus $\sum_{u \in V(G')} g(u) \geq \gamma_{hR}(G') = |V(G^{'})|$. Now

suppose that $g(a) = 0$. Note that in this case either a is hop Roman dominate by a vertex of $V(G^{'}) - \{a\}$ or a is hop Roman dominate by v_2. If a is hop Roman dominate by a vertex of $V(G^{'}) - \{a\}$, then clearly $g|_{V(G^{'})}$ is a HRDF for $G^{'}$. Thus $\sum_{u \in V(G^{'})} g(u) \geq \gamma_{hR}(G') = |V(G^{'})|$. Next suppose that a is hop Roman dominate by v_2. Note that in this case $g(v_2) = 2$, $g(v) = 0$, $g(v_1) = 1$. Then we can change $g(v)$ to 2, $g(a)$ to 1, $g(v_1)$ and $g(v_2)$ to 0 to obtain a $\gamma_{hR}(G)-$function, $g^{'}$, such that $g^{'}(v) + g^{'}(v_1) + g^{'}(v_2) \geq 2$ and $g^{'}|_{V(G^{'})}$ be a HRDF for $G^{'}$. Then $\sum_{u \in V(G^{'})} g^{'}(u) \geq \gamma_{hR}(G') = |V(G^{'})|$.

Next suppose that any vertex of $V(P_2)$ is adjacent to exactly $|V(G^{'})| - 1$ vertices of $G^{'}$. First suppose that a be a vertex of $G^{'}$ such that $a \notin N(v_1) \cup N(v_2)$. Clearly $d(a, v_1) = d(a, v_2) = 2$. Then from Operation $\mathcal{O}_2$, we conclude that a is adjacent to any vertex of $V(G^{'}) - \{a\}$. Let g be a $\gamma_{hR}(G)-$function. If $g(v) + g(v_1) + g(v_2) \leq 1$, then clearly $g(v) = 1$ and two vertices v_1 and v_2 are hop Roman dominated by a. Thus, $g(a) = 2$. Then we can change $g(a)$ to 1 and $g(v)$ to 2 to obtain a $\gamma_{hR}(G)-$function, $g^{'}$, with $g^{'}(v) + g^{'}(v_1) + g^{'}(v_2) \geq 2$ such that $g^{'}|_{V(G')}$ be a HRDF for $G^{'}$. Then $\sum_{u \in V(G^{'})} g^{'}(u) \geq \gamma_{hR}(G') = |V(G^{'})|$.

Now suppose that a, b be two vertices of $V(G^{'})$ such that $a \neq b$ and $a \notin N(v_1)$, $b \notin N(v_2)$. Clearly, $d(a, v_1) = d(b, v_2) = 2$. Note that in this case, from Operation $\mathcal{O}_2$, we conclude that a and b are adjacent to any vertex of $V(G^{'}) - \{a, b\}$ and note that in this case $1 \leq d(a, b) \leq 2$. Let g be a $\gamma_{hR}(G)-$function. If $g(v) + g(v_1) + g(v_2) \leq 1$, then clearly $g(v) = 1$ and $g(a) = g(b) = 2$. Then we can change $g(a)$ and $g(b)$ to 1 and $g(v)$ to 2, to obtain a HRDF for G with weight less than g, a contradiction. Thus $g(v) + g(v_1) + g(v_2) \geq 2$. Now suppose that $\sum_{u \in V(G^{'})} g(u) < |V(G^{'})|$. Then, there is $u \in V(G^{'})$, such that $g(u) = 0$ and u is hop Roman dominated just by one vertex of $\{v_1, v_2\}$. Note that in this case $u = a$ or $u = b$. If $g(a) = g(b) = 0$, then $g(v_1) = g(v_2) = 2$ and clearly $g(v) = 0$. Then we can change $g(v)$ to 2, $g(v_1)$ and $g(v_2)$ to 0, and also $g(a)$ and $g(b)$ to 1, to obtain a $\gamma_{hR}(G)-$function, $g^{'}$, such that $g^{'}(v) + g^{'}(v_1) + g^{'}(v_2) \geq 2$ and also $g^{'}|_{V(G^{'})}$ be a HRDF for $G^{'}$. Then $\sum_{u \in V(G^{'})} g^{'}(u) \geq \gamma_{hR}(G') = |V(G^{'})|$. Now suppose that $g(a) = 0$ and $g(b) \neq 0$. Then, clearly $g(v_1) = 2$ and $g(v) = 0$. Suppose

that $g(v_2) = 0$, then $g(b) = 2$. Thus we can change $g(v)$ to 2, $g(v_1)$ and $g(v_2)$ to 0, and also $g(a)$ and $g(b)$ to 1, to obtain a $\gamma_{hR}(G)-$function, $g^{'}$, such that $g^{'}(v) + g^{'}(v_1) + g^{'}(v_2) \geq 2$ and also $g^{'}|_{V(G^{'})}$ be a HRDF for $G^{'}$. Then $\sum_{u \in V(G^{'})} g^{'}(u) \geq \gamma_{hR}(G') = |V(G^{'})|$. Next suppose that $g(v_2) \neq 0$. Note that in this case $g(v_2) = 1$. Then we can change $g(v)$ to 2, $g(v_1)$ and $g(v_2)$ to 0, and also $g(a)$ to 1, to obtain a $\gamma_{hR}(G)-$function, $g^{'}$, such that $g^{'}(v) + g^{'}(v_1) + g^{'}(v_2) \geq 2$ and also $g^{'}|_{V(G^{'})}$ be a HRDF for $G^{'}$. Then $\sum_{u \in V(G^{'})} g^{'}(u) \geq \gamma_{hR}(G') = |V(G^{'})|$. ♦

Now from Claim 1 we conclude that $|V(G)| - 1 = 2 + |V(G^{'})| \leq \gamma_{hR}(G)$. Therefore, $\gamma_{hR}(G) = |V(G)| - 1$. ■

Let $\mathcal{F}$ be the family of graphs illustrated in Figure 1. We are now ready to state the main result of this section.

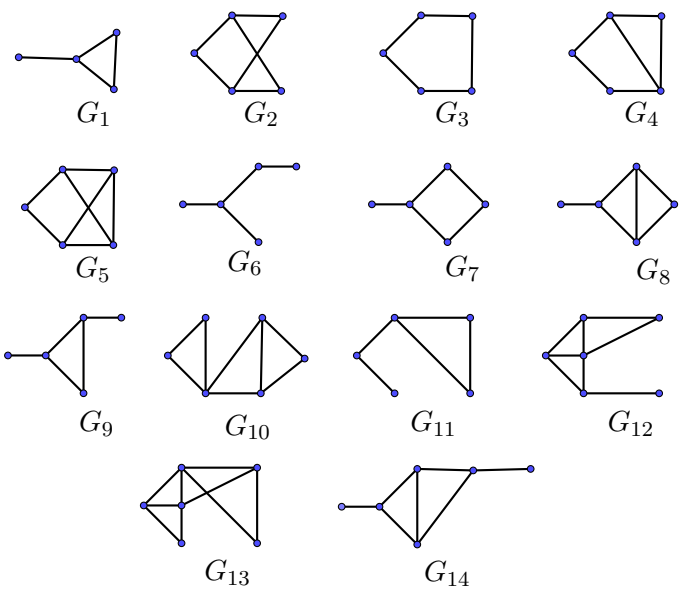

Figure 1. All graphs in the family $\mathcal{F}$

Theorem 5 *If G is a connected graph of order n, then $\gamma_{hR}(G) = n-1$ if and only if $G \in \mathcal{G}^* \cup \mathcal{F} \cup \{P_5\}$.*

Proof. ($\Rightarrow$) Let G be a connected graph of order n with $\gamma_{hR}(G) = n-1$. If $\Delta_h(G) \geq 3$, then Proposition 2 leads to $\gamma_{hR}(G) \leq n-2$, a contradiction. Thus $\Delta_h(G) \leq 2$. If $\Delta_h(G) = 1$, then Propositions 1 and 2 imply that $\gamma_{hR}(G) = n$, a contradiction. Thus $\Delta_h(G) = 2$. Let v be a vertex of G with $\deg_h(v) = \Delta_h(G) = 2$ and $N_2(v) = \{v_1, v_2\}$ and let $G' = G[N(v)]$. We proceed with two claims namely Claim 1 and Claim 2.

Claim 1. If $\deg_h(z) = \deg_h(w) = 2$ for two distinct vertices z and w of G, then $N_2(z) \cap N_2(w) \neq \emptyset$.

Proof of Claim 1: Assume that z, w be two distinct vertices of G such that $\deg_h(z) = \deg_h(w) = 2$. Suppose to the contrary that $N_2(z) \cap N_2(w) = \emptyset$. Then f defined by $f(z) = f(w) = 2$, $f(x) = 0$ for $x \in N_2(z) \cup N_2(w)$, and $f(u) = 1$ otherwise, is a HRDF for G with weight $n-2$. Thus $\gamma_{hR}(G) \leq n-2$, a contradiction. ♦

Claim 2. $\delta(G') \geq |V(G')| - 2$.

Proof of Claim 2: If there are three vertices $a, b, c \in V(G')$ such that $a \notin N(b) \cup N(c)$, then $\deg_h(a) \geq 2$. Thus $\deg_h(a) = 2$ and $N_2(a) = \{b, c\}$. Since $\deg_h(a) = \deg_h(v) = 2$ and $N_2(a) \cap N_2(v) = \emptyset$, then Claim 1 leads to a contradiction. Therefore, any vertex of G' is adjacent to at least $|V(G')| - 2$ vertices of G'. Therefore, $\delta(G') \geq |V(G')| - 2$. ♦

Note that $diam(G) \geq 2$, since $\gamma_{hR}(G) = n-1$. Suppose that $diam(G) \geq 5$. Let $P : u_1u_2u_3u_4u_5u_6...u_d$ be diametrical path in G. Then $u_1, u_5 \in N_2(u_3)$ and $u_2, u_6 \in N_2(u_4)$. Since $\Delta_h(G) = 2$, thus $\deg_h(u_3) = \deg_h(u_4) = 2$. This contradicts Claim 1. Thus $diam(G) \leq 4$. Therefore, we conclude that $2 \leq diam(G) \leq 4$. We consider the following cases.

Case 1. $diam(G) = 2$. Clearly $V(G) - \big(N_2(v) \cup N[v]\big) = \emptyset$. According to the Claim 2, we consider two following subcases.

Subcase 1.1. $\delta(G') = |V(G')| - 1$. Then G' is a complete graph. Assume that $v_1v_2 \notin E(G)$. Note that $d(v_1, v_2) = 2$. If $|V(G^{'})| = 1$, then any vertex of $N_2(v)$ is adjacent to the vertex of $G^{'}$, and so G is obtained from $G^{'}$ by Operation $\mathcal{O}_1$. Thus assume that $|V(G^{'})| \geq 2$. If

at least a vertex of $N_2(v)$ is adjacent to at most $|V(G')| - 1$ vertices of $G^{'}$, then $\Delta_h(G) = 2$ leads to a contradiction. Thus both vertices v_1 and v_2 are adjacent to any vertex of $V(G')$. Hence G is obtained from $G^{'}$ by Operation $\mathcal{O}_1$. Next assume that $v_1v_2 \in E(G)$. If $|V(G')| = 1$, then v_1 and v_2 are adjacent to the vertex of $G^{'}$. So $G = G_1 \in \mathcal{F}$. Thus assume that $|V(G')| \geq 2$. Then we can see that there exists at most one vertex in $N(v)$, such as x, such that $d(v_1, x) = 2$. Also there exists at most one vertex in $N(v)$, such as y, such that $d(v_2, y) = 2$. Thus v_1 and v_2 are adjacent to at least $|V(G')| - 1$ vertices of G'. Therefore, G is obtained from G' by Operation $\mathcal{O}_2$. Therefore, $G \in \mathcal{G}^*$.

Subcase 1.2. $\delta(G') = |V(G')| - 2$. Clearly $|V(G')| \geq 2$. Suppose that $|V(G')| \geq 3$. First we show that $G' \in \mathcal{G}$. Suppose to the contrary, that $\gamma_{hR}(G') \leq |V(G')| - 1$. Let f be a $\gamma_{hR}(G')$-function. Then we can define g with $g(v) = 2$, $g(v_1) = g(v_2) = 0$ and $g(z) = f(z)$ otherwise, to obtain a HRDF for graph G with weight $g(V) = f(V) + 2$. Thus $\gamma_{hR}(G) \leq f(V) + 2 \leq |V(G')| - 1 + 2 = |V(G')| + 1$. Since $|V(G)| = |V(G')| + 3$, therefore we have $\gamma_{hR}(G) \leq |V(G)| - 2 = n - 2$, a contradiction. Therefore, $\gamma_{hR}(G') = |V(G')|$ and by Theorem 3, $G' \in \mathcal{G} \cup \{P_4, K_{n-3}\}$. Since $\delta(G') = |V(G')| - 2$, thus $G' \neq P_4$ and $G' \neq K_{n-3}$. Therefore, $G' \in \mathcal{G}$. Assume that $v_1v_2 \notin E(G)$. Then there exists at most one vertex in $N(v) \cup \{v_2\}$ at distance two from v_1 (since otherwise $\deg_h(v_1) > 2$, a contradiction). Since $diam(G) = 2$, we have $d(v_1, v_2) = 2$. Then we conclude that v_1 is adjacent to any vertex of $G^{'}$. Similarly v_2 is adjacent to any vertex of $G^{'}$. Therefore, G is obtained from $G^{'}$ by using Operation $\mathcal{O}_1$. Therefore, $G \in \mathcal{G}^*$. Now suppose that $v_1v_2 \in E(G)$. Note that in this case, if there is $z \in V(G^{'})$, such that $z \notin N(v_1)$, then $d(z, v_1) = 2$. Similarly if $z \notin N(v_2)$, then $d(z, v_2) = 2$. Thus v_1 and v_2 are adjacent to at least $|V(G^{'})| - 1$ vertices of $G^{'}$. Let $u \in V(G')$ be a vertex with $\deg_{G'}(u) = \delta(G') = |V(G')| - 2$ and let w be the vertex of G' such that $w \notin N(u)$. Clearly $d_{G'}(u, w) = 2$. Then $u, w \in N(v_1) \cup N(v_2)$ (since in the otherwise, if $u \notin N(v_1) \cup N(v_2)$ or $w \notin N(v_1) \cup N(v_2)$, then $\deg_h(u) > 2$ or $\deg_h(w) > 2$, a contradiction). Now suppose that $u \notin N(v_1) \cap N(v_2)$. Then since $u \in N(v_1) \cup N(v_2)$, so either $u \in N(v_1)$ or $u \in N(v_2)$. Without loss of generality, assume that $u \in N(v_1)$ and $u \notin N(v_2)$. Note that in this case v_1 is adjacent

to any vertex of $N_{G'}(u)$ (since otherwise, $\deg_h(v_1) = \deg_h(u) = 2$, $N_2(v_1) \cap N_2(u) = \emptyset$, and Claim 1 leads to contradiction). Also v_2 is adjacent to any vertex of $N_{G'}(u)$, since otherwise $\deg_h(v_2) > 2$, a contradiction. Therefore, G is obtained from $G^{'} \in \mathcal{G}$ by operation $\mathcal{O}_2$. Therefore, $G \in \mathcal{G}^*$.

Now suppose that $|V(G')| = 2$. Then $G^{'} = \overline{K_2}$. If $v_1v_2 \notin E(G)$, then v_1 and v_2 are adjacent to any vertex of G', since otherwise there exist two vertices at distance three or four from each other, a contradiction with $diam(G) = 2$. Thus $G = G_2 \in \mathcal{F}$. Thus assume that $v_1v_2 \in E(G)$. Note that in this case any vertex of $V(G')$ is adjacent to at least one vertex of $N_2(v)$, since otherwise there exist two vertices in G at distance three from each other, a contradiction with $diam(G) = 2$. On the other hand any vertex of $N_2(v)$ is adjacent to at least one vertex of $V(G^{'})$. We conclude that $G \in \{G_3, G_4, G_5\}$. Consequently, $G \in \mathcal{F}$.

Case 2. $diam(G) = 3$. According to the Claim 2, we consider two following subcases.

Subcase 2.1. $\delta(G') = |V(G')| - 1$. Then G' is a complete graph. Assume that $|V(G')| \geq 3$. We can easily see that v_1 and v_2 are adjacent to at least $|V(G')| - 1$ vertices of G' and $N(v_1) \cap N(v_2) \neq \emptyset$. Let $v_1v_2 \notin E(G)$, then $d(v_1, v_2) = 2$. If there is $z \in N(v)$ such that $z \notin N(v_1)$, then $z \in N_2(v_1)$. Thus $\deg_h(v_1) \geq 3$, a contradiction. Therefore, v_1 is adjacent to any vertex of $N(v)$. Similarly we can show that v_2 is adjacent to any vertex of $N(v)$. In this case, distance between any two vertices of $N[v] \cup N_2(v)$ is at most two. Since $diam(G) = 3$, thus we may suppose that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$. Note that in this case there is a vertex z in $V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Then clearly $\deg_h(z) \geq 3$, a contradiction. Next suppose that $v_1v_2 \in E(G)$. Note that in this case v_1 and v_2 are adjacent to at least $|V(G')| - 1$ vertices of G'. Thus distance between any two vertices of $N[v] \cup N_2(v)$ is at most two. Since $diam(G) = 3$, thus we may suppose that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$. Note that in this case there is a vertex z in $V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Then clearly by Claim 1 or this fact that $\Delta_h(G) = 2$, we have a contradiction. Hence, we conclude that $|V(G')| \leq 2$.

Suppose that $|V(G')| = 1$. Let $V(G') = \{w\}$. Then v_1 and v_2

are adjacent to w and distance between any two vertices of $N[v] \cup N_2(v)$ is at most two. Since $diam(G) = 3$, thus we may suppose that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$. Let z be a vertex of $V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Then suppose that there is a vertex $z' \in V(G) - (N[v] \cup N_2(v))$ such that $z' \neq z$. If $z' \in N(v_1) \cup N(v_2)$, then $z, z' \in N_2(w)$ and by this fact that $\Delta_h(G) = 2$ we have $\deg_h(w) = 2$. Then Claim 1 leads to a contradiction. Hence, $z' \in N(z) - \{v_1, v_2\}$. Then clearly $d(v, z') = 4$, a contradiction. Thus $|V(G) - (N[v] \cup N_2(v))| \leq 1$. Since $diam(G) = 3$, we conclude that $|V(G) - (N[v] \cup N_2(v))| = 1$. Now suppose that $v_1v_2 \notin E(G)$, then we can easily see that $G \in \{G_6, G_7\}$ and if $v_1v_2 \in E(G)$, then we can easily see that $G \in \{G_8, G_9\}$. Therefore, $G \in \mathcal{F}$.

Next suppose that $|V(G')| = 2$. Then $G' = K_2$. Suppose that $v_1v_2 \notin E(G)$. If a vertex of $N_2(v)$ is adjacent to both vertices of G', then the other vertex of $N_2(v)$ is also adjacent to both vertices of G' (otherwise this fact that $\Delta_h(G) = 2$ leads to contradiction). Note that in this case for any pair $a, b \in N[v] \cup N_2(v)$, $d(a, b) \leq 2$. Since $diam(G) = 3$, thus we may suppose that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$. Let z be a vertex of $V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Then $\deg_h(z) \geq 2$. Since $\Delta_h(G) = 2$, we have $\deg_h(z) = 2$. Note that $N_2(z) = N(v)$, that Claim 1 leads to a contradiction. Hence any vertex of $N_2(v)$ is adjacent to one vertex of G'. If $N_{G'}(v_1) \cap N_{G'}(v_2) \neq \emptyset$, then $\deg_h(v_1) = \deg_h(v_2) > 2$, a contradiction. Thus $N_{G'}(v_1) \cap N_{G'}(v_2) = \emptyset$ and $d(v_1, v_2) = 3$. If $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$, then Claim 1 or this fact that $diam(G) = 3$, leads to contradiction. Therefore, we conclude that $V(G) - (N[v] \cup N_2(v)) = \emptyset$. Therefore, $G = G_9 \in \mathcal{F}$.

Now suppose that $v_1v_2 \in E(G)$. If at least one vertex of $N_2(v)$ is adjacent to both vertices of $N(v)$, then for any pair $a, b \in N[v] \cup N_2(v)$, $d(a, b) \leq 2$. Since $diam(G) = 3$, thus we may suppose that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$. Let z be a vertex of $V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Then $\deg_h(z) \geq 2$. Since $\Delta_h(G) = 2$, we have $\deg_h(z) = 2$. Note that in this case, Claim 1 leads to contradiction. Hence each vertex of $N_2(v)$ is adjacent to one vertex of $V(G')$. If $N_{G'}(v_1) \cap N_{G'}(v_2) = \emptyset$, then for any pair $a, b \in N[v] \cup N_2(v)$, $d(a, b) \leq 2$. Thus we may suppose that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$. Let z be a vertex

of $V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Then clearly $\deg_h(z) = 2$ and we can see that Claim 1 leads to a contradiction. Hence $N_{G'}(v_1) \cap N_{G'}(v_2) \neq \emptyset$. Let $\{x\} = N_{G'}(v_1) \cap N_{G'}(v_2)$. Clearly for any pair $a, b \in N[v] \cup N_2(v)$, $d(a, b) \leq 2$. Since $diam(G) = 3$, thus we may suppose that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$. Let z be a vertex of $V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Suppose that there is $z^{'} \in V(G) - (N[v] \cup N_2(v))$, such that $z \neq z^{'}$. If $z^{'} \in N(v_1) \cup N(v_2)$, then $\deg_h(x) = 2$ and $N_2(x) \cap N_2(v) = \emptyset$, and so Claim 1 leads to contradiction. Hence we shall have $z^{'} \in N(z) - \{v_1, v_2\}$, that in this case clearly $d(z^{'}, v) = 4$, a contradiction. Thus $|V(G) - (N[v] \cup N_2(v))| \leq 1$. If z is adjacent to only one vertex of $N_2(v)$, without loss of generality, suppose that z be only adjacent to v_1, then $\deg_h(v_2) = 3$, a contradiction. Hence $z \in N(v_1) \cap N(v_2)$ and $G = G_{10} \in \mathcal{F}$.

Subcase 2.2. $\delta(G') = |V(G')| - 2$. Clearly $|V(G')| \geq 2$. Suppose that $|V(G')| \geq 4$. Let x be a vertex of $G^{'}$ such that $\deg_{G'}(x) = |V(G^{'})| - 2$ and let y be the other vertex of $G^{'}$ such that $y \notin N_{G'}(x)$. We can see that $d(x, y) = 2$ and $N_{G'}(x) = N_{G'}(y)$. We show that if there is a vertex $a \in V(G^{'})$ such that $a \notin N(v_1)$, then $d(a, v_1) = 2$. If $a = x$, then at least one vertex of $N_{G'}(x)$ is adjacent to the vertex v_1, since in the otherwise, $\deg_h(v_1) > 2$, a contradiction. Thus we deduce that $d(a, v_1) = 2$. Next suppose that $a \in N_{G'}(x)$. If $\{x, y\} \cap N(v_1) = \emptyset$, then $\deg_h(v_1) > 2$, a contradiction. Thus $\{x, y\} \cap N(v_1) \neq \emptyset$, and so $d(a, v_1) = 2$. If there are two vertices $a, b \in V(G^{'})$ such that $a, b \notin N(v_1)$. Then, $\deg_h(v_1) > 2$, a contradiction. Therefore, we conclude that v_1 and v_2 are adjacent to at least $|V(G')| - 1$ vertices of G' and clearly $N_{G'}(v_1) \cap N_{G'}(v_2) \neq \emptyset$. Note that, we have $d(v_1, v_2) \leq 2$. Then clearly distance between any two vertices of $N[v] \cup N_2(v)$ is at most two. Since $diam(G) = 3$, thus we may suppose that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$. Let z be a vertex of $V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Then clearly $\deg_h(z) > 2$, a contradiction. Consequently, $|V(G')| \leq 3$.

Now suppose that $|V(G')| = 2$. Then $G' = \overline{K_2}$. Let $V(G') = \{x, y\}$. Let $v_1v_2 \notin E(G)$. If v_1 and v_2 are adjacent to both vertices of $V(G')$, then distance between any two vertices of $N[v] \cup N_2(v)$ is at most two. Since $diam(G) = 3$, thus we may suppose that $V(G) - (N[v] \cup N_2(v)) \neq$

$\emptyset$. Let z be a vertex of $V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Then clearly $\deg_h(z) = 2$ and $N_2(z) \cap N_2(v) = \emptyset$. By Claim 1 we have a contradiction. Thus there exists at least one vertex in $N_2(v)$ such that is adjacent to only one vertex of G'. If v_1 is adjacent to both vertices of $V(G')$ and v_2 be adjacent to only one vertex of $V(G')$, then $d(v_2, x) = 3$ or $d(v_2, y) = 3$. We show that $V(G) - (N[v] \cup N_2(v)) = \emptyset$. Suppose to contrary that there is $z \in V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Then we can see that Claim 1 or $diam(G) = 3$ leads to contradiction. Therefore, we conclude that $G = G_7 \in \mathcal{F}$. Now suppose that each vertex of $N_2(v)$ is adjacent to one vertex of G'. If v_1 is adjacent to x and v_2 be adjacent to y, then $d(v_1, v_2) = 4$, a contradiction. Hence, both vertices of $N_2(v)$ are adjacent to precisely one vertex of G', without loss of generality, suppose that v_1 and v_2 are adjacent to x. Clearly $d(y, v_1) = d(y, v_2) = 3$. Then we can easily see that $V(G) - (N[v] \cup N_2(v)) = \emptyset$. Thus we deduce that $G = G_6 \in \mathcal{F}$.

Now suppose that $v_1v_2 \in E(G)$. If there is at least a vertex in $N_2(v)$ such that is adjacent to both vertices of G', then distance between any two vertices of $N[v] \cup N_2(v)$ is at most two. Since $diam(G) = 3$, we may suppose that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$. Let z be a vertex of $V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Then clearly $\deg_h(z) = 2$ and Claim 1 leads to contradiction. Therefore, we conclude that each vertex of $N_2(v)$ is adjacent to one vertex of G'. If v_1 is adjacent to x and v_2 be adjacent to y, then distance between any two vertices of $N[v] \cup N_2(v)$ is at most two and we can easily see that $V(G) - (N[v] \cup N_2(v)) = \emptyset$. Thus both v_1 and v_2 are adjacent to precisely one vertex of G'. In this case, there exist two vertices in $N[v] \cup N_2(v))$ such that distance between them is equal to three and also we can easily see that $V(G) - (N[v] \cup N_2(v)) = \emptyset$. Therefore $G = G_{11} \in \mathcal{F}$.

Next Suppose that $|V(G')| = 3$. Then G' is a path of length two. Let G' be the path $P : xyu$. Assume that $v_1v_2 \notin E(G)$. If one of the vertices $N_2(v)$ is adjacent to all vertices of $G^{'}$, then clearly the other vertex of $N_2(v)$ is also adjacent to all vertices of $G^{'}$. Now suppose that both vertices of $N_2(v)$ are adjacent to any vertex of $V(G^{'})$, then distance between any two vertices of $N[v] \cup N_2(v)$ is at most two. Since

$diam(G) = 3$, thus we may suppose that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$. Let z be a vertex of $V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Then clearly $\deg_h(z) > 2$, a contradiction. Thus each vertex of $N_2(v)$ is adjacent to at most two vertices of G'. If each vertex of $N_2(v)$ is adjacent to two vertices of G', then clearly $N_{G'}(v_1) \cap N_{G'}(v_2) \neq \emptyset$ and so $d(v_1, v_2) = 2$. Thus hop-degree of each vertex of $N_2(v)$ is more than two, a contradiction. Thus at least one vertex of $N_2(v)$ is adjacent to only one vertex of G'. Suppose that both vertices of $N_2(v)$ are adjacent to one vertex of G', then we can easily see that $y \notin N(v_1) \cup N(v_2)$, and also $N_{G'}(v_1) \cap N_{G'}(v_2) = \emptyset$. Thus if v_1 is adjacent to x, then v_2 is adjacent to u. Then $d(v_1, v_2) = 4$, a contradiction with $diam(G) = 3$. Thus one vertex of $N_2(v)$ is adjacent to one vertex of G' and the other vertex is adjacent to two vertices of G'. Without loss of generality suppose that v_1 is adjacent to two vertices of G' and v_2 is adjacent to only one vertex of G'. Then clearly $N_{G'}(v_1) \cap N_{G'}(v_2) = \emptyset$, and as before, we observe that v_2 is not adjacent to y. Note that in this case $d(v_1, v_2) = 3$ and we can easily see that $V(G) - (N[v] \cup N_2(v)) = \emptyset$. Therefore, we conclude that $G = G_{12} \in \mathcal{F}$.

Now assume that $v_1v_2 \in E(G)$. If any vertex of $N_2(v)$ is adjacent to at least two vertices of G', then distance between any two vertices of $N[v] \cup N_2(v)$ is at most two. Since $diam(G) = 3$, thus we may suppose that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$. Let z be a vertex of $V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Then $\deg_h(z) > 2$, that is a contradiction. Hence at least one vertex of $N_2(v)$ is adjacent to only one vertex of G'. Note that in this case, any vertex of $N_2(v)$ is adjacent to at most two vertices of G'. If both vertices v_1 and v_2 are adjacent to one vertex of G', then $y \notin N(v_1) \cup N(v_2)$, since in the otherwise $\deg_h(v_1) \geq 3$ and $\deg_h(v_2) \geq 3$, a contradiction. Also $N_{G'}(v_1) \cap N_{G'}(v_2) \neq \emptyset$, in the otherwise $\deg_h(v_1) > 2$ and $\deg_h(v_2) > 2$. Hence v_1 and v_2 are adjacent to precisely one vertex of $\{x, u\}$ and so $G = G_{10} \in \mathcal{F}$. Now suppose that a vertex of $N_2(v)$ be adjacent to both vertices of G' and the other vertex of $N_2(v)$ is adjacent to one vertex of G'. Without loss of generality suppose that v_2 is adjacent to one vertex of G'. Then $y \notin N(v_2)$, since in the otherwise $\deg_h(v_2) \geq 3$, a contradiction. Without loss of generality suppose that v_2 is adjacent to u. If $u \notin N(v_1)$, then $\{x, y\} =$

$N_{G'}(v_1)$ and we can see that $\deg_h(v_2) \geq 3$, a contradiction. Therefore, $u \in N(v_1)$. Next suppose that $x \in N(v_1)$, then we can easily see that $\deg_h(v_2) > 2$, a contradiction. Hence, we have $N_{G'}(v_1) = \{y, u\}$ and $N_{G'}(v_2) = \{u\}$. Note that in this case $d(x, v_2) = 3$ and we can see that $V(G) - (N[v] \cup N_2(v)) = \emptyset$. Consequently, we have $G = G_{13} \in \mathcal{F}$.

Case 3. $diam(G) = 4$. According to the Claim 2, we consider two following subcases.

Subcase 3.1. $\delta(G') = |V(G')| - 1$. Then G' is a complete graph. Suppose that $|V(G')| \geq 3$. Clearly each vertex of $N_2(v)$ is adjacent to at least $|V(G')| - 1$ vertices of G'. Also v_1 is adjacent to at least $|V(G') \cup \{v_2\}| - 1$ vertices of $V(G') \cup \{v_2\}$, since in the otherwise if there exist two vertices $a, b \in V(G') \cup \{v_2\}$ such that $a, b \notin N(v_1)$, then $\deg_h(v_1) > 2$, a contradiction. Similarly, we can see that v_2 is adjacent to at least $|V(G') \cup \{v_1\}| - 1$ vertices of $V(G') \cup \{v_1\}$. Clearly, in this case distance between any two vertices of $N[v] \cup N_2(v)$ is at most two. Since $diam(G) = 4$, we conclude that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$. Then there is the vertex $z \in V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Note that in this case $\deg_h(z) \geq 2$ and the fact that $\Delta_h(G) = 2$ or Claim 1, lead to contradiction. Therefore, we may assume that $|V(G')| \leq 2$.

We first assume that $|V(G')| = 2$, then we can easily see that distance between any two vertices of $N[v] \cup N_2(v)$ is at most three. Since $diam(G) = 4$, we may suppose that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$. Note that in this case from the fact that $\Delta_h(G) = 2$ or Claim 1 we have a contradiction.

Hence we can assume that $|V(G')| = 1$. Let $V(G') = \{z\}$. Then v_1 and v_2 are adjacent to z and distance between every two vertices in $N[v] \cup N_2(v)$ is at most two. Since $diam(G) = 4$, we conclude that there is a vertex $y \in V(G) - (N[v] \cup N_2(v))$, such that $d(y, v_1) = 2$ or $d(y, v_2) = 2$. In this case, if $v_1v_2 \notin E(G)$, then hop-degree of at least one vertex of $N_2(v)$ is more than two, a contradiction. Thus $v_1v_2 \in E(G)$. Now suppose that $y \in V(G) - (N[v] \cup N_2(v))$ is a vertex at distance two from v_1 and let v_1xy be the path between v_1 and y. If $x \notin N(v_2)$, then $\deg_h(x) = \deg_h(v_1) = 2$ and $N_2(x) \cap N_2(v_1) = \emptyset$, that is a contradiction with Claim 1. Thus $x \in N(v_2)$. If there exists a vertex $x' \in V(G) - (N[v] \cup N_2(v))$ such that $x' \neq x$ and $x' \in N(v_2) \cup N(v_1)$,

then $\deg_h(z) = 2$, and $N_2(z) \cap N_2(v) = \emptyset$, thus Claim 1 leads to contradiction. Since $y \in V(G) - (N[v] \cup N_2(v))$ and $y \neq x$, so we conclude that $y \notin N(v_2)$ and since $diam(G) = 4$, so $N(y) - \{x\} = \emptyset$ and y is a leaf. On the other hand $N(x) - N_2(v) = \{y\}$, since in the otherwise, $\deg_h(v_1) \geq 3$ and also $\deg_h(v_2) \geq 3$, a contradiction. Thus we conclude that $G = G_{14} \in \mathcal{F}$. Similarly, we can show that if $y \in V(G) - (N[v] \cup N_2(v))$ is a vertex at distance two from v_2, then $G = G_{14} \in \mathcal{F}$.

Subcase 3.2. $\delta(G') = |V(G')| - 2$. Clearly $|V(G')| \geq 2$. Suppose that $|V(G')| \geq 4$. Let x be a vertex of $V(G^{'})$ such that $\deg_{G'}(x) = |V(G^{'})| - 2$ and let y be a vertex of $G^{'}$ such that $y \notin N_{G'}(x)$. We can see that $d(x, y) = 2$ and $N_{G'}(x) = N_{G'}(y)$. We show that if there is a vertex $a \in V(G^{'})$ such that $a \notin N(v_1)$, then $d(a, v_1) = 2$. If $a = x$, then at least one vertex of $N_{G'}(x)$ is adjacent to the vertex v_1, since in the otherwise, $\deg_h(v_1) > 2$, a contradiction. Thus we deduce that $d(a, v_1) = 2$. Next suppose that $a \in N_{G'}(x)$. If $\{x, y\} \cap N(v_1) = \emptyset$, then $\deg_h(v_1) > 2$, a contradiction. Thus $\{x, y\} \cap N(v_1) \neq \emptyset$, and so $d(a, v_1) = 2$. If there are two vertices $a, b \in V(G^{'})$ such that $a, b \notin N(v_1)$, then $\deg_h(v_1) > 2$, a contradiction. Therefore, we conclude that v_1 is adjacent to at least $|V(G')| - 1$ vertices of G'. With a similar argument we can show that v_2 is adjacent to least $|V(G')| - 1$ vertices of G'. Clearly $N_{G'}(v_1) \cap N_{G'}(v_2) \neq \emptyset$. Then the distance between any two vertices of $N[v] \cup N_2(v)$ is at most 2. We may suppose that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$, then there is the vertex $z \in V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Note that in this case clearly $\deg_h(z) > 2$, a contradiction. Hence, we conclude that $V(G) - (N[v] \cup N_2(v)) = \emptyset$ and so we may assume that $|V(G')| \leq 3$.

We first assume that $|V(G')| = 3$. Then G' is a path of length two. Let G' be the path $P : xyu$. Assume that at least one vertex of $N_2(v)$ is adjacent to at least two vertices of G'. Without loss of generality, suppose that v_1 is adjacent to at least two vertices of G'. Then clearly the distance between any two vertices of $N[v] \cup N_2(v)$ is at most 3. Thus we may suppose that there is at least a vertex $z \in V(G) - (N[v] \cup N_2(v))$ that is adjacent to v_1 or v_2. If z is adjacent to v_1, then $\deg_h(z) \geq 2$ and Claim 1 or the fact that $\Delta_h(G) = 2$

leads to a contradiction. Thus z is adjacent to v_2. Note that in this case v_2 is adjacent to at least one vertex of G'. Then we can easily see that Claim 1 or the fact that $\Delta_h(G) = 2$ leads to a contradiction. Therefore, each vertex of $N_2(v)$ is adjacent to one vertex of G'. We can see that $y \notin N(v_1) \cup N(v_2)$. If $N_{G'}(v_1) \cap N_{G'}(v_2) \neq \emptyset$, then the distance between any two vertices of $N[v] \cup N_2(v)$ is at most 3. Thus we may suppose that there is a vertex $z \in V(G) - (N[v] \cup N_2(v))$ that is adjacent to v_1 or v_2. Let $x \in N_{G'}(v_1) \cap N_{G'}(v_2)$. Then $\deg_h(x) = 2$ and $N_2(x) = \{u, z\}$. Thus $N_2(x) \cap N_2(v) = \emptyset$ and so Claim 1 leads to a contradiction. Thus $N_{G'}(v_1) \cap N_{G'}(v_2) = \emptyset$. If $v_1v_2 \in E(G)$, then clearly $\deg_h(v_1) > 2$ and $\deg_h(v_2) > 2$, a contradiction. Thus $v_1v_2 \notin E(G)$. Then $d(v_1, v_2) = 4$ and clearly $V(G) - (N[v] \cup N_2(v)) = \emptyset$, and so we conclude that $G = G_{14} \in \mathcal{F}$.

Next we assume that $|V(G')| = 2$, then $G' = \overline{K_2}$. Let $V(G') = \{x, y\}$. Clearly the vertices v_1 and v_2 are adjacent to at least one vertex of G'. If each vertex of $N_2(v)$ is adjacent to both vertices of G', then clearly the distance between any two vertices in $N[v] \cup N_2(v)$ is at most two. Since $diam(G) = 4$, we may suppose that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$. Then there is the vertex $z \in V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Note that in this case $\deg_h(z) \geq 2$ and Claim 1 or the fact that $\Delta_h(G) = 2$ lead to a contradiction. Thus there is at least one vertex in $N_2(v)$ such that it is adjacent to only one vertex of G'. Suppose that one vertex of $N_2(v)$ is adjacent to both vertices of $V(G')$ and the other vertex is adjacent to only one vertex of G'. Then the distance between any two vertices of $N[v] \cup N_2(v)$ is at most two. Since $diam(G) = 4$, we may suppose that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$. Then there is the vertex $z \in V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Note that in this case $\deg_h(z) \geq 2$ and Claim 1 or the fact that $\Delta_h(G) = 2$ lead to a contradiction. Therefore, any vertex of $N_2(v)$ is adjacent to only one vertex of G'. Assume that $v_1v_2 \in E(G)$. Note that in this case the distance between any two vertices of $N[v] \cup N_2(v)$ is at most 3. We may suppose that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$. Then there is the vertex $z \in V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Then we can easily see that Claim 1 or the fact that $\Delta_h(G) = 2$ lead to a contradiction. Hence $v_1v_2 \notin E(G)$. Assume $N_{G'}(v_1) \cap N_{G'}(v_2) \neq \emptyset$,

then clearly the distance between any two vertices of $N[v] \cup N_2(v)$ is at most 3. Thus we may suppose that $V(G) - (N[v] \cup N_2(v)) \neq \emptyset$. Then there is the vertex $z \in V(G) - (N[v] \cup N_2(v))$ such that $z \in N(v_1) \cup N(v_2)$. Note that in this case we can easily see that Claim 1 leads to a contradiction. Therefore, $N_{G'}(v_1) \cap N_{G'}(v_2) = \emptyset$, and $d(v_1, v_2) = 4$. Thus we conclude that $V(G) - (N[v] \cup N_2(v)) = \emptyset$ and so $G = P_5$.

($\Leftarrow$) Suppose that $G \in \mathcal{G}^* \cup \mathcal{F} \cup \{P_5\}$. It is obvious that if $G = P_5$ or $G \in \mathcal{F}$, then $\gamma_{hR}(G) = n - 1$. Next suppose that $G \in \mathcal{G}^*$. Then by Lemma 4, we have $\gamma_{hR}(G) = n - 1$. ■

References

[1] M. Adabi, E. Ebrahimi Targhi, N. Jafari Rad, and M. Saied Moradi, "Properties of independent Roman domination in graphs," *Australas. J. Combin.*, vol. 52, pp. 11–18, 2012.

[2] R. A. Beeler, T. W. Haynes, and S. T. Hedetniemi, "Double Roman domination," *Discrete Appl. Math.*, vol. 211, pp. 23–29, 2016.

[3] S. K. Ayyaswamy, B. Krishnakumari, C. Natarajan, and Y. B. Venkatakrishnan, "Bounds On The Hop Domination Number Of a Tree," *Proc. Indian Acad. Sci. (Math. Sci.)*, vol. 125, no. 4, pp. 449–455, 2015.

[4] S. K. Ayyaswamy and C. Natarajan, *Hop Domination in Graphs.*

[5] E.J. Cockayane, P.M. Dreyer Jr., S.M. Hedetniemi, and S.T. Hedetniemi, "On Roman domination in graphs," *Discrete Math.*, vol. 278, pp. 11–22, 2004.

[6] M. Farhadi Jalalvand and N. Jafari Rad, "On the complexity of k-step and k-hop dominating sets in graphs," *Math. Montisnigri*, vol. 40, pp. 36–41, 2017.

[7] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, *Fundamentals of Domination in Graphs*, New York: Marcel Dekker Inc., 1998.

[8] M. A. Henning and N. Jafari Rad, "On 2-Step and Hop Dominating Sets in Graphs," *Graphs Combin.*, vol. 33, no. 4, pp. 913–927, 2017.

[9] C. Natarajan and S. K. Ayyaswamy, "Hop Domination in Graphs-II," *An. Stt. Univ. Ovidius Const.* vol. 23, no. 2, pp. 187–199, 2015.

[10] N. J. Rad and H. Rahbani, "Bounds on the locating Roman dominating number in trees," *Discuss. Math. Graph Theory*, vol. 38, no. 1, pp. 49–62, 2018.

[11] C. S. ReVelle and K. E. Rosing, "Defendens imperium Romanum: a classical problem in military strategy," *Amer Math. Monthly*, vol. 107, pp. 585–594, 2000.

[12] E. Shabani, *Hop Roman domination in graphs*, 2017.

[13] I. Stewart, "Defend the roman empire!," *Sci. Amer.*, vol. 281, no. 6, pp. 136–139, 1999.

E. Shabani[1], N. Jafari Rad[1], A. Poureidi[1],

Received September 17, 2018
Revised December 17, 2018

[1] School of Mathematical Sciences,
Shahrood University of Technology,
Shahrood, Iran
E–mail: n.jafarirad@gmail.com

Computer Science Journal of Moldova, vol.**27**, no.1(79), 2019

Connected Dominating Sets and a New Graph Invariant

Vladimir Bercov

Abstract

Based on concept of connected dominating sets of a simple graph G we introduce a new invariant $\eta(G)$ which does not exceed the number of Hadwiger. The Nordhaus-Gaddum inequalities are: $\eta(G)\eta(\overline{G}) \geq n(G)$ and $\eta(G) + \eta(\overline{G}) \leq 6n(G)/5$. For values of chromatic number $\chi(G) \leq 4$ we prove $\eta(G) \geq \chi(G)$. We put forward the hypothesis: the last inequality holds for all simple graphs G.

Keywords: dominating set, number of Hadwiger, chromatic number, Nordhaus-Gaddum inequalities.

1 Introduction

All graphs G considered in this paper are undirected, simple and finite with vertex set $V(G)$. We denote $|V(G)|$ by $n(G)$. For $X \subseteq V(G)$, we denote by $G[X]$ the subgraph of G induced by X, further, $G - X = G[V(G) - X]$. For the subgraph H of graph G, $G-H = G[V(G)-V(H)]$. We shall write $v \sim u$ ($v \nsim u$) when vertices v and u are (are not) adjacent. If every pair of vertices in X are adjacent, then $G[X]$ is a complete subgraph or a clique K. The clique number $\omega(G)$ of a graph G is the number of vertices in a maximum clique in G. The degree of a vertex is the number of edges incident to the vertex. The number of Hadwiger $h(G)$ is the largest number of connected subgraphs of G, pairwise without common vertices and connected with at least one edge. Contracting the edges within each of these subgraphs so that each subgraph collapses to a single vertex produces a

maximum complete graph $K_{h(G)}$ on $h(G)$ vertices. From definition of $h(G)$ it follows that $h(G) = max_{1 \leq i \leq k} h(G_i)$, where $G_1, G_2, \ldots, G_k$ are components of disconnected graph G.

If $D \subseteq V(G)$, $G[D]$ is connected and every vertex not in D has a neighbor in D, then it's clear that $h(G-D) \leq h(G)-1$. This property allows us to introduce new invariants of the graph, in the definitions of which we use the concept of connected dominating sets, and these invariants do not exceed the number of Hadwiger.

2 Domination and New Invariants

Definition 1 *A connected dominating set D is a subset of vertices of a graph G such that every vertex is in D or adjacent to at least one vertex in D, and $G[D]$ is connected in all connected components (subgraphs) of G.*

Further, unless otherwise specified, dominating set means connected dominating set. The edge $e = vu$ is dominating if the set $\{v, u\}$ is dominating.

Definition 2 *Let V_1 be a dominating set in a graph G, let V_2 be a dominating set in $G - V_1$, V_3 – dominating set in $G - V_1 - V_2$ and so on. $\eta_0(G)$ is the maximum length of the sequence of dominating sets $V_1, V_2, V_3, \ldots$.*

If $V_1, V_2, \ldots, V_{\eta_0}$ is a maximum sequence of dominating sets, then it's obvious that $|V_{\eta_0}| = 1$ or the set V_{η_0} is independent, i.e. no two of its vertices are adjacent.

Theorem 1 *For any graph G*

(i) $\eta_0(G) \leq h(G)$,

(ii) If D is any dominating set of G, then $\eta_0(G) \geq \eta_0(G - D) + 1$,

(iii) $\eta_0(G) = max_{1 \leq j \leq k} \eta_0(G_j)$, where $G_1, G_2, \ldots, G_k$ are connected components of G.

Proof. Let $\eta_0(G) = \eta_0$.

(i) If we contract all edges in induced subgraphs $G(V_1), G(V_2), \ldots, G(V_{\eta_0}))$, we obtain a graph G^c with clique number $\omega(G^c) = \eta_0$. From definition of number of Hadwiger, $h(G) \geq \omega(G^c)$.

(ii) If $V_1, V_2, \ldots, V_l$ is a maximum sequence of dominating sets of $G - D$, then $D, V_1, V_2, \ldots, V_l$ is a sequence of dominating sets of G.

(iii) Let $\eta_{0j} = \eta_0(G_j)$ for all $j (1 \leq j \leq k)$ and $V_1^j, V_2^j, \ldots, V_{\eta_{0j}}^j$ be a maximum sequence of dominating sets in G_j. Let $max_{1 \leq j \leq k} \eta_{0j} = \eta_{01}$. If for some j, $\eta_{0j} < \eta_{01}$, then we put $V_i^j = \varnothing$ for $\eta_{0j} + 1 \leq i \leq \eta_{01}$. The sequence $V_1, V_2, \ldots, V_{\eta_{01}}$, where $V_i = \cup_{j=1}^k V_i^j$, is the sequence of dominating sets in G. So, $\eta_0(G) \geq max_{1 \leq j \leq k} \eta_0(G_j)$.

Let $V_1, V_2, \ldots, V_{\eta_0}$ be a maximum sequence of dominating sets in graph G. For all $1 \leq i \leq \eta_0$, $V_i = \cup_{j=1}^k V_i^j$, where V_i^j is a corresponding dominating set in G_j or $V_i^j = \varnothing$. If for $i = i_0$, $V_{i_0}^j \neq \varnothing$, then $V_i^j \neq \varnothing$ for all $1 \leq i \leq i_0$. Since $V_{\eta_0} \neq \varnothing$, there exists at least one j for which the sequence $V_1^j, V_2^j, \ldots, V_{\eta_0}^j$ is the sequence of dominating sets in G_j. So, $\eta_0(G) \leq max_{1 \leq j \leq k} \eta_0(G_j)$.

Therefore, $\eta_0(G) = max_{1 \leq j \leq k} \eta_0(G_j)$. ■

The independence number $\alpha(G)$ of a graph G is the size of the largest independent set of its vertices. Duchet and Meyniel [2] showed that for every connected graph G with independence number $\alpha(G)$, there exists a dominating set with at most $2\alpha(G) - 1$ vertices, and therefore $h(G)(2\alpha(G) - 1) \geq n(G)$.

Theorem 2 *Let G be any graph with k connected components. Then*

$$\eta_0(G)(2\alpha(G) - k) \geq n(G).$$

Proof. The proof is by induction on $n = n(G)$. For $n \leq 3$, the result is true by inspection. Suppose $n \geq 4$ and suppose the result is true for all graphs with fewer than n vertices, and let G be a graph with $n(G) = n$ and its components are $G_1, G_2, \ldots, G_k$. In each component G_i $(1 \leq i \leq k)$ there exists dominating set D_i and $|D_i| \leq 2\alpha(G_i) - 1$. If $D = \cup_{i=1}^k D_i$, then $|D| = \sum_{i=1}^k |D_i| \leq \sum_{i=1}^k (2\alpha(G_i) - 1) = 2\alpha(G) - k$. The induction hypotheses implies that $\eta_0(G)(2\alpha(G) - k) =$

$\eta_0(G)\sum_{i=1}^{k}(2\alpha(G_i)-1) \geq \sum_{i=1}^{k}\eta_0(G_i)(2\alpha(G_i)-1) \geq \sum_{i=1}^{k}(\eta_0(G_i - D_i)+1)(2\alpha(G_i)-1) = \sum_{i=1}^{k}\eta_0(G_i-D_i)(2\alpha(G_i)-1)+\sum_{i=1}^{k}(2\alpha(G_i)-1) \geq \sum_{i=1}^{k}\eta_0(G_i-D_i)(2\alpha(G_i-D_i)-1)+2\alpha(G)-k \geq \sum_{i=1}^{k}n(G_i-D_i)+2\alpha(G)-k = \sum_{i=1}^{k}n(G_i)-\sum_{i=1}^{k}|D_i|+2\alpha(G)-k \geq \sum_{i=1}^{k}n(G_i) = n(G)$. ■

Let $C_l = \{v_1, v_2, \ldots, v_l\}$ be a chordless cycle. A Mycielsky graph M_k of order k is a triangle-free graph ($\omega(M_k) = 2$) with chromatic number $\chi(M_k) = k$ (see [5]). M_2 contains two connected vertices, $M_3 = C_5$. If $V(M_k) = v_1, v_2, \ldots, v_n$, the graph M_{k+1} contains M_k itself as a subgraph, together with $n+1$ vertices: $u_1, u_2, \ldots, u_n, w$. Each vertex u_i is connected to w, and for each edge v_iv_j of M_k, graph M_{k+1} includes two edges, u_iv_j and u_jv_i. The set $\{u_1, u_2, \ldots, u_n, w\}$ is dominating in M_{k+1} and $\eta_0(M_{k+1}) \geq \eta_0(M_k)+1$. Because $\eta_0(M_2) = 2$, $\eta_0(M_3) = 3$, for any order k: $\eta_0(M_k) \geq \chi(M_k)$. So, the difference $\eta_0(G) - \omega(G)$ can be arbitrarily large. However, if G is obtained by adding some adjacent vertices with degree one to each vertex of any graph H, then $\eta_0(G) = 2$.

Definition 3 *$\eta(G) = max\ \eta_0(G')$, where the maximum is taken over all induced subgraphs G' of the graph G.*

Theorem 3 *For any graph G*

(i) $\omega(G) \leq \eta(G) \leq h(G)$,

(ii) If D is any dominating set in G then $\eta(G) \geq \eta(G-D)+1$,

(iii) If $\triangle(G)$ the maximum degree, then $\eta(G) \leq \triangle(G)+1$.

Proof. *(i)* Let $\omega = \omega(G)$. For any maximum clique K_ω, $\eta_0(K_\omega) = \omega$, so $\eta(G) \geq \omega$. If G' is one of subgraphs for which $\eta_0(G') = \eta(G)$, then, by Theorem 2, $\eta_0(G') \leq h(G')$, and therefore, $\eta(G) \leq h(G)$.

(ii) D is dominating set for all subgraphs of G containing D. The statement follows from Theorem 2.

(iii) If $\eta = \eta(G)$ and $V_1, V_2, \ldots, V_\eta$ is one of the longest sequence of dominating sets, then V_η is an independent set. Degrees of all vertices in this set are at least $\eta - 1$. ■

If we take n cycles C_{n-1}, join every two different cycles with a single edge using different vertices in each cycle, then for obtained graph G with $\triangle(G) = 3$, $h(G) \geq n$, $\eta(G) \leq 4$. So, the difference $h(G) - \eta(G)$ can be arbitrarily large.

The new invariant $\eta(G)$ also can be defined as a maximum length of a sequence of subsets of vertices $U_1, U_2, U_3, \ldots$, where $U_i \cap U_j = \varnothing$ $(i \neq j)$, every vertex from $U_1 \cup U_2 \cup \ldots \cup U_{k-1}$ is adjacent to at least one vertex in U_k, and subgraph induced by U_k $(k \geq 2)$ is connected.

In the proof of some lower bounds for number of Hadwiger we can just replace h by η. Below we give a proof of one theorem (see [6]) with this replacement.

Theorem 4 *Let G be any graph with $\alpha(G) = 2$. Let $n = n(G)$, $\omega = \omega(G)$, then $\eta(G) \geq (n + \omega)/3$.*

Proof. We proceed to prove by induction on n. Let $K = K_\omega$ be a maximum clique of G and let $H = G - V(K)$. By a 2-path of H we mean an induced subpath of length 2. The vertex set of any 2-path is dominating in G. If H contains a 2-path P, then $\omega(G-P) = \omega(G) = \omega$ and the induction hypothesis implies that $\eta(G) \geq \eta(G - P) + 1 \geq \frac{(n-3+\omega)}{3} + 1 = (n + \omega)/3$. If H does not contain any 2-path, then H is either a complete graph or disjoint union of two complete graphs. In both cases we claim that $\omega \geq n/2$. In the first case, this is evident. In the second case, H is a disjoint union of two complete graphs, say H_1 and H_2 and, because of $\alpha(G) = 2$, every vertex of the complete subgraph K is either joined to all vertices of H_1 or to all vertices of H_2. This implies the claim. Consequently, $\eta(G) \geq \omega \geq (n + \omega)/3$. Thus Theorem 4 is proved. ■

3 The Nordhaus-Gaddum inequalities for $\eta(G)$

Nordhaus and Gaddum studied the chromatic number in a graph G and its complement $\overline{G}$ together. They proved lower and upper bounds of the sum and of the product of chromatic numbers of G and $\overline{G}$ in terms of $n(G)$. Since then, any bound of the sum and/or the product of an

invariant in a graph and its complement is called a Nordhaus-Gaddum type inequality. We prove these inequalities for $\eta(G)$.

The diameter of a connected graph G, denoted $d(G)$, is the maximum distance between two vertices. If graph is not connected, then the diameter is defined as infinite. Clearly, if $d(G) \geq 3$, then in its complement $\overline{G}$ exists dominating edge. Let G be connected. We call the induced subgraph G_0 a cut-subgraph if $G - G_0$ is disconnected. If $d(G) = 2$, then the vertex set of any connected cut-subgraph is dominating in G.

Lemma 1 *If G is connected, not complete and does not have a complete cut-subgraph, then $h(G) \geq \omega(G) + 1$.*

Proof. Let K_ω be a maximum clique in G and subgraph $H = G - K_\omega$ is connected. If we assume that H is not joined to some vertex $v \in V(K_\omega)$, then $K_\omega - v$ is complete cut-subgraph, hence $V(H)$ is dominating in G. ■

Lemma 2 *If K is a complete subgraph of G and $G - K$ is connected, then in G exists dominating set D such that $|D| \leq 2\alpha(G) - 1$ and $|D \cap V(K)| \leq 1$.*

Proof. Let $H = G - K$ and $v \sim u$, where $v \in V(K)$ and $u \in V(H)$. By the result of Duchet and Meyniel (see [2]), the subgraph H has the dominating set D_1 with $p + q$ vertices; p vertices form independent set (independent vertices), and $q \leq p - 1$. If $p = \alpha(G)$, then $D = D_1$. If $p \leq \alpha(G) - 1$ and D_1 is not dominating in G, then $D = D_1 \cup \{v, u\}$. In this case $|D| = |D_1| + 2 = p + q + 2 \leq 2p - 1 + 2 \leq 2(\alpha(G) - 1) + 1 = 2\alpha(G) - 1$. ■

Theorem 5 $\eta(G) \cdot \eta(\overline{G}) \geq n(G)$.

Proof. We proceed by induction on $n = n(G)$. For $n \leq 5$, the result is clear. Suppose $n \geq 6$, and statement holds for all graphs with fewer than n vertices. Clearly, if $max\{d(G), d(\overline{G})\} \geq 3$, then Theorem 5 holds. Let $d(G) = d(\overline{G}) = 2$.

Case 1: A complete cut-subgraph K exists in graph G (or in $\overline{G}$) and $H_1, H_2, \ldots, H_k$ ($k \geq 2$) are connected components of $G - K$. In complement $\overline{G}$ (or in G) each edge $e = v_i v_j$ ($v_i \in V(H_i)$, $v_j \in V(H_j)$, $i \neq j$) is dominating in subgraph induced in $\overline{G}$ by the set of vertices $V(G) - V(K)$. Let $G' = G - K - \{v_i, v_j\}$. We have: $\eta(G) \cdot \eta(\overline{G}) \geq \eta(G)(\eta(\overline{G'}) + 1) = \eta(G)\eta(\overline{G'}) + \eta(G) \geq (\eta(G') + 1)\eta(\overline{G'}) + \eta(G) = \eta(G')\eta(\overline{G'}) + \eta(\overline{G'}) + \eta(G) \geq n(G') + \eta(\overline{G'}) + \eta(G) = n(G) - n(K) - 2 + \eta(\overline{G'}) + \eta(G) \geq n(G) + \eta(\overline{G'}) - 2$. If $\eta(\overline{G'}) \leq 1$, then $\omega(G) \geq (n-2)/2$, hence Theorem 5 holds in this case.

Case 2: Neither G nor $\overline{G}$ does not contain a complete cut-subgraph. Without loss of generality we may assume $\omega = \omega(G) \geq \alpha(G) = \alpha$. Let K_ω be a maximum clique of G. By Lemma 2, G has a dominating set D with $n(D) \leq 2\alpha(G) - 1$ and $|D \cap V(K_\omega)| \leq 1$.

Case 2.1: D is a dominating set in G, $|D| = p + 1$. The set D consists of p independent vertices and one vertex adjacent to all p vertices. Let $G' = G - D$. Thus $\eta(G)\eta(\overline{G}) \geq (\eta(G') + 1)\eta(\overline{G}) = \eta(G')\eta(\overline{G}) + \eta(\overline{G}) \geq \eta(G')\eta(\overline{G'}) + \eta(\overline{G}) \geq n - p - 1 + \eta(\overline{G})$. By Lemma 1, $\eta(\overline{G}) \geq \omega(\overline{G}) + 1 \geq p + 1$, hence Theorem 5 holds in this case.

Case 2.2: D is a dominating set in G, $|D| = p + q$, p vertices are independent and $2 \leq q \leq p - 1$. None of the vertices of G is joined to all p independent vertices. In this case these p vertices are dominating in $\overline{G}$. Let $G' = G - D$. We have: $\eta(G)\eta(\overline{G}) \geq (\eta(G') + 1)\eta(\overline{G}) = \eta(G')\eta(\overline{G}) + \eta(\overline{G}) \geq \eta(G')(\eta(\overline{G'}) + 1) + \eta(\overline{G}) = \eta(G')\eta(\overline{G'}) + \eta(G') + \eta(\overline{G}) \geq n(G') + \eta(G') + \eta(\overline{G}) = n(G) - p - q + \eta(G') + \eta(\overline{G}) \geq n(G) - 2\alpha + 1 + \eta(G') + \eta(\overline{G}) \geq n(G) - 2\alpha + 1 + \omega - 1 + \alpha \geq n(G) - \alpha + \omega \geq n(G)$.

■

Now we prove the upper bound of a sum.

Kostochka proved (see [4]) that for number of Hadwiger, $h(G) + h(\overline{G}) \leq 6n(G)/5$. We show that there are graphs G for which $\eta(G) + \eta(\overline{G}) = 6n(G)/5$.

Given a graph G, we say that graph $H = inf(G)$ is an inflation of G if each vertex $v \in V(G)$ is replaced by a complete graph K^v. If $u \in V(G)$ and $v \sim u$, then in H every vertex of K^v is joined to every vertex of K^u. We call the complete graphs K^v, K^u the atoms of inflation.

Let $H = inf(C_5)$ and for corresponding atoms $n(K^1) = n(K^2) = n(K^3) = n(K^4) = n(K^5) = k$. In complete K^2 and K^4 we remove all edges and the resulting graph we denote by G. In the graph G, subgraph induced by the set of vertices $V(K^1) \cup V(K^5)$ is complete. In the complement $\overline{G}$, subgraph induced by the set of vertices $V(K^2) \cup V(K^4)$ is complete, hence $\omega(G) = \omega(\overline{G}) = 2k$. There are k disjoint dominating sets $\{v_2, v_3, v_4\}$ in $G : v_2 \in V(K^2)$, $v_3 \in V(K^3)$, $v_4 \in V(K^4)$, and k disjoint dominating sets $\{v_1, v_3, v_5\}$ in $\overline{G} : v_1 \in V(K^1)$, $v_3 \in V(K^3)$, $v_5 \in V(K^5)$. So $\eta(G) \geq 3k$, $\eta(\overline{G}) \geq 3k$, and $\eta(G)+\eta(\overline{G}) \geq 6k = 6n(G)/5$.

4 New Invariant and Chromatic Number

Let G be any graph and chromatic number $\chi(G) = \chi$. In definition of chromatic number, for independent sets of vertices $V_1, V_2, , V_\chi, \cup_{i=1}^{\chi} V_i = V(G)$, $V_i \cap V_j = \varnothing$ $(i \neq j)$, we can always assume that V_1 is a dominating set in graph G, V_2 is a dominating set in $G - V_1$, V_3 is a dominating set in $G - V_1 - V_2$ and so on.

Now we begin to study the relationship between the chromatic number and $\eta(G)$.

Let $\delta(G)$ be the minimum degree.

Lemma 3 *If $\delta(G) \geq 3$, then in G exists a cycle and connected subgraph joined to all vertices of this cycle.*

Proof. We define the distance between two cycles C^1 and C^2 as the shortest distance between all pairs of vertices v and u, where $v \in C^1$ and $u \in C^2$. If two cycles have common vertices, then the distance between them is zero. Among the pairs of cycles with maximum distance ρ, choose one C_l with minimum length l. If $G - C_l$ is connected, then

C_l and $G - C_l$ are desired. Let $G - C_l$ be not connected. If $\rho = 0$, then all components of $G - C_l$ do not have cycles. If $\rho > 0$, then only one component has cycles (otherwise ρ is not the maximum). Let T be any component without cycles. Consider the graph $H = C_l \cup T$. First, prove that $l \leq 4$.

An arbitrary terminal vertex $t \in V(T)$ is joined to at least two vertices of C_l. These two vertices divide C_l into two simple paths. If $l \geq 5$, then the length of one path is at least 3, and then the length of the cycle induced by vertices of another path and t is less than l.

If T is joined to all vertices of C_l, then C_l and T are desired. Assume that vertex $v_1 \in C_l$ is not joined to T. From the minimality of l it follows that the number of terminal vertices in T is at least two.

Case 1. $l = 3$ and $C_3 = \{v_1, v_2, v_3\}$. In the graph T all terminal vertices $t_1, t_2, \ldots, t_s$ $(s \geq 2)$ are joined to v_2 and to v_3. The cycle $C = \{t_1, v_2, v_3\}$ and the graph $T - t_1$ are desired.

Case 2. $l = 4$ and $C_4 = \{v_1, v_2, v_3, v_4\}$. All terminal vertices $t_1, t_2, \ldots, t_s$ $(s \geq 2)$ are adjacent to v_2 and to v_4. Assume that $s = 2$, i.e. T is the chain and the number of vertices $n(T) \geq 3$. If in this chain $t_1 \sim t$, then $t \sim v_3$. The cycle $C = \{t_1, v_2, t_2, v_4\}$ and the induced subgraph with the set of vertices $V(T) \cup \{v_3\} - \{t_1, t_2\}$ are desired. Now assume that $s \geq 3$. The cycle C is the same as above and the connected graph is induced by vertices $V(T) - \{t_1, t_2\}$.

■

Theorem 6 *If $\chi(G) \leq 4$, then $\eta(G) \geq \chi(G)$.*

Proof. The cases where $\chi(G) = 1, 2$ are trivial. The case $\chi(G) = 3$ is also easy: the graph requiring three colors has an odd cycle $C_l = \{v_1, v_2, \ldots, v_l\}$. The set of vertices $\{v_3, v_4, \ldots, v_l\}$ is joined to connected v_1 and v_2. If $\chi(G) = 4$, then without loss of generality, we may assume that G is critical, i.e. for each vertex v: $\chi(G - v) = 3$. For such graphs (see [1]) $\delta(G) \geq 3$, and by Lemma 3, $\eta(G) \geq 4$. ■

Let $\chi(k) = max\{\chi(G) \mid \eta(G) \leq k\}$.

Theorem 7 $\chi(k) \leq 3 \cdot 2^{k-3}$ *for* $k \geq 3$.

Proof. We proceed by induction on k. As it is shown above, $\chi(3) = 3$. So suppose $k \geq 4$ and suppose the result is true for all graphs with $\eta < k$ and let $G = (V, E)$ be a graph with $\eta(G) = k$. Let $v_0 \in V$ and D_i be subgraph induced by all vertices at distance i from v_0. Subgraph $G[\{v_0\} \cup D_1 \cup D_2 \cup \ldots \cup D_{i-1}]$ is connected, so $\eta(D_i) \leq k-1$ for all i. Clearly, that $\chi(G) \leq max\{\chi(D_{i-1}) + \chi(D_i)\}$ $(i = 1, 2, 3, \ldots)$, where $D_0 = \{v_0\}$. So, $\chi(G) \leq 2 \cdot \chi(k-1) \leq 2 \cdot 3 \cdot 2^{k-1-3} = 3 \cdot 2^{k-3}$. ■

Conjecture: For all graphs G, $\chi(G) \leq \eta(G)$.

5 Concluding Remarks

The Hadwigers conjecture (HC) can be stated in form: For all graphs G, $\chi(G) \leq h(G)$. Therefore, a new conjecture is a strengthening of HC. If HC is false, then a counterexample might possibly be obtained among counterexamples to the new conjecture.

A good place to look for a counterexample to our conjecture are graphs G with independence number $\alpha(G) = 2$. For such graphs $\chi(G) \geq n(G)/2$. In [3] Kim has proved that there is a constant $c > 0$ such that there exist graphs G on n vertices, with $\alpha(G) = 2$ and clique number $\omega(G) \leq c\sqrt{n \cdot \log n}$. For these graphs to have $\chi(G) \leq \eta(G)$ we need to find a sequence with at least $\chi(G) - c\sqrt{n \cdot \log n}$ dominating sets with at least two vertices.

References

[1] G.A. Dirak, "A property of 4-chromatic graphs and some remarks on critical graphs," *J. London Math. Soc.*, vol. s1-27, no. 1, pp. 85–92, 1952.

[2] P. Duchet and H. Meyniel, "On Hadwiger's Number and the Stability Number," in *Graph Theory* (North-Holland Mathematical Studies, vol. 62), Béla Bollobás, Ed. Amsterdam: North-

Holland, 1982, pp. 71-73. ISSN: 0304-0208. DOI: 10.1016/S0304-0208(08)73549-7.

[3] J.H. Kim, "The Ramsey number $R(3,t)$ has order of magnitude $t^2/\log t$," *Random Struct. Algorithms*, vol. 7, pp. 173–207, 1995.

[4] A.V. Kostochka, "On Hadwiger number of a graph and its complement," in *Finite and Infinite Sets* (Colloq. Math. Soc. János Bolyai, vol. 37), 1984, pp. 537–545.

[5] Jan Mycielski, "Sur Le Coloriage des Graphes," *Colloquium Mathematicae*, vol. 3, no. 2, pp. 161–162, 1955. (in French)

[6] M. Plummer, M. Stiebitz, and B. Toft, "On a special case of Hadwigers conjecture," *Discuss. Math. Graph Theory*, vol. 23, pp. 333-363, 2003.

Vladimir Bercov,

Department of Mathematics
BMCC CUNY
New York, USA
E–mail: vbercov@bmcc.cuny.edu

Received June 21, 2018
Revised January 16, 2019

Computer Science Journal of Moldova, vol.**27**, no.1(79), 2019

Error correcting codes from sub-exceeding functions

L. Rabefihavanana, H. Andriatahiny, T. Rabeherimanana

Abstract

In this paper, we present linear systematic error-correcting codes $\mathcal{L}_k$ and $\mathcal{L}_k^+$ which are the results of our research on the sub-exceeding functions.

Given an integer k such that $k \geq 3$, these two codes are respectively $[2k, k]$ and $[3k, k]$ linear codes. The minimum distance of $\mathcal{L}_3$ is 3 and for $k \geq 4$ the minimum distance of $\mathcal{L}_k$ is 4. The code $\mathcal{L}_k^+$, the minimum distances are respectively 5 and 6 for $k = 4$ and $k \geq 5$.

By calculating the complexity of the algorithms, our codes have fast and efficient decoding.

Then, for a short and medium distance data transmission (wifi network, bluetooth, cable, ...), we see that the codes mentioned above present many advantages.

Keywords: Error correction code, encoding, decoding, sub-exceeding function.

MSC 2010: 94B05; 94B35; 11B34.

1 Introduction

New information and communication technologies or NICTs require today a norm increasingly strict in terms of quality of service. The diversity and the increasing volumes of data exchanged/processed also require increasingly fast and reliable systems.

In these constraints related to information processing, we need to take into account the increased sensitivity of technologies in front of

external disruptive sources. It's about especially to protect information against environmental damage during transmission.

The aim of this article is to build new error correcting codes using the results of our two articles entitled: *Parts of a set and sub-exceeding function: coding and decoding* [16] in 2017 and *Encoding of Partition Set Using Sub-exceeding Function* [15] in 2018.

In the last section of this article, we give the decoding algorithm of these codes using Groebner basis.

2 Preliminaries

Let n be a positive integer, $[\![n]\!]$ denotes the set of positive integers less or equal to n and the zero element, i.e.

$$[\![n]\!] = \{0, 1,\ 2,\ ...\ n\}\,.$$

2.1 The necessary ones on the study of sub-exceeding functions

Definition 2.1. (See [16]) Let n be a positive integer and let f be a map from $[n]$ to $[\![n]\!]$. This function f is said sub-exceeding if for all i in $[\![n]\!]$, we have

$$f(i) \leq i.$$

We denote by $\mathcal{F}_n$ the set of all sub-exceeding functions on $[n]$, i.e.

$$\mathcal{F}_n = \{f :\ [\![n]\!] \longrightarrow [\![n]\!] \ \mid\ f(i) \leq i,\ \forall i \in [\![n]\!]\}\,. \tag{1}$$

Remark 2.2. A sub-exceeding function f can be represented by the word of $n+1$ alphabet $f(0)f(1)f(2)...f(n)$. So, we describe f by its images $f = f(0)f(1)f(2)...f(n)$.

Definition 2.3. (See [16]) Let n and k be two integers such that $0 \leq k \leq n$. We define by $\mathcal{H}_n^k$ the subset of $\mathcal{F}_n$ such that

$$\mathcal{H}_n^k = \{f \in \mathcal{F}_n \mid f(i) \leqslant f(i+1) \text{ for all } i \in [\![n]\!] \text{ and } Im(f) = [\![k]\!]\}\ . \tag{2}$$

Here, $\mathcal{H}_n^k$ is the set of all sub-exceeding functions of $\mathcal{F}_n$ with a quasi-increasing sequence of images formed by all elements of $[\![k]\!]$.

Example 2.4. *Take $n = 4$ and $k = 3$. We find that the function $f = 01123$ is really in $\mathcal{H}_4^3$ because $(f(i))_{0\leq i\leq 4}$ is a quasi-increasing sequence formed by all the elements of $[\![3]\!]$. But if we take $f = 01133$, even if the sequence $(f(i))_{0\leq i\leq 4}$ is quasi-increasing, $f = 01133 \notin \mathcal{H}_4^3$ because $Im(f) \neq [\![3]\!]$ (without 2 among the $f(i)$).*

Following Definition 2.3, we denote by $\mathcal{H}_n$ the set defined as follows:

$$\mathcal{H}_n = \bigcup_{k=0}^{n} \mathcal{H}_n^k. \tag{3}$$

Theorem 2.5. *(See [16])*
Let n and k be two integers such that $0 \leq k \leq n$.

1. *For $k = 0$, we always find that $\mathcal{H}_n^0$ is a set of singletons:*

$$\mathcal{H}_n^0 \ = \ \{f \ = \ 000...00_{n+1\text{-terms}}\}.$$

2. *For $k = n$, we also find that $\mathcal{H}_n^n$ is a set of singletons:*

$$\mathcal{H}_n^n \ = \ \{f \ = \ 0123....(n-1)(n)\}.$$

3. *For any integer k such that $0 < k < n$, we can construct all sub-exceeding functions of $\mathcal{H}_n^k$ as follows:*

   (a) *Take all the elements of $\mathcal{H}_{n-1}^{k-1}$ and add the integer k at the end,*

   (b) *Take all the elements of $\mathcal{H}_{n-1}^k$ and add the integer k at the end*

   To better presentation of this construction, we adopt the following writing:

$$\mathcal{H}_n^k \ = \ \left\{\mathcal{H}_{n-1}^{k-1} \curvearrowleft k\right\} \bigcup \left\{\mathcal{H}_{n-1}^k \curvearrowleft k\right\}.$$

   Here, $(*) \curvearrowleft k$ means that we add the integer k at the end of all elements of $(*)$.

Table 1. The iteration table of the elements of $\mathcal{H}_n^k$

$n \quad \diagdown \quad k$	0	1	2	3	4	...
0	0					
1	00	01				
2	000	001 011	012			
3	0000	0001 0011 0111	0012 0112 0122	0123		
4	00000	00001 00011 00111 01111	00012 00112 01112 00122 01122 01222	00123 01123 01223 01233	01234	

From Theorem 2.5, we have this Table 1 which presents all elements of $\mathcal{H}_n^k$ for some integers n ($n = 0,\ 1,\ 2,\ 3$ and 4).

Proposition 2.6. *See [16]*
Let n and k be two integers such that $0 \leqslant k \leqslant n$. So, we have the following relations:

1. *Card* $\mathcal{H}_n^0 = Card\ \mathcal{H}_n^n = 1$,
2. *Card* $\mathcal{H}_n^k = Card\ \mathcal{H}_{n-1}^{k-1} + Card\ \mathcal{H}_{n-1}^k$,
3. *Card* $\mathcal{H}_n^k = \binom{n}{k}$ *and Card* $\mathcal{H}_n = 2^n$.

Proof. From the construction of the elements of $\mathcal{H}_n^k$ in Theorem 2.5, we have directly the result of Proposition 2.6. □

This Proposition 2.6 presents to us the iterative calculus of the cardinal of $\mathcal{H}_n^k$.

Thus, Table 2 below gives the cardinal of $\mathcal{H}_n^k$ for some integers n ($n = 0,\ 1,\ 2,\ 3$ and 4).

Table 2. The cardinal table of $\mathcal{H}_n^k$

$n \quad \diagdown \quad k$	0	1	2	3	4	...
0	1					
1	1	1				
2	1	2	1			
3	1	3	3	1		
4	1	4	6	4	1	
$\vdots$						

Thus constructed, Table 2 is none other than the Pascal triangle.

3 Main result: error-correcting codes from the study on the sub-exceeding function

In this section, we present our linear error-correcting code from sub-exceeding function.

3.1 The error-correcting code constructions

Recall that for a positive integer n, a function f from $[\![n]\!]$ to $[\![n]\!]$ is said to be sub-exceeding if for any integer i in $[\![n]\!]$, we always have the inequality $f(i) \leq i$.

Thus, the sub-exceeding term amounts to saying that the image of an integer i by an application f is always an integer smaller or equal to this one.

Theorem 3.1. *Let k be a positive integer and let f be an application from $[\![k]\!]$ to $\mathbb{F}_2^{k+1}$. Then the application f is a sub-exceeding function if and only if $f(0) = 0$.*

This theorem tells us that all message of k bits on $\mathbb{F}_2$ which begins with 0 is a sub-exceeding function.

Proof. We say that the image of an integer i in $[\![k]\!]$ by the application f is always equal to 0 or 1. Thus, by the condition $f(0) = 0$, we have $f(i) \leq i$ for all i. So, f is a sub-exceeding function. □

Now, let's examine the subset $\mathcal{H}_k$ for the set of sub-exceeding functions in all application from $[\![k]\!]$ in $\mathbb{F}_2^{k+1}$. That is to say the subset $\mathcal{H}_k$ for the set of k bits messages on $\mathbb{F}_2$.

Referring to Theorem 2.5, we can have all the elements of $\mathcal{H}_k$ (see Table 3). Moreover, from Proposition 2.6, we find

$$\text{Card } (\mathcal{H}_k^0) = 1 \text{ and that } \text{Card}(\mathcal{H}_k^1) = k. \tag{4}$$

Table 3 shows the elements of $\mathcal{H}_k^i$ for each value of $i \in \{0, 1\}$ and some integer k.

Definition 3.2. For a positive integer k, we define by T_k the matrix of $k+1$ rows and k columns such that

$$T_k = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & 1 \\ 0 & 0 & 0 & \dots & 1 & 1 & 0 \\ \vdots & \vdots & \vdots & \nearrow & \nearrow & \vdots & \vdots \\ 0 & 0 & 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & 1 & 1 & \dots & 1 \end{pmatrix}. \tag{5}$$

Here, $T[i, j]$ denotes the element of T_k in the i^{th} row and the j^{th} column and

- for all i in $\{1,\ 2,\ ..,\ k\}$,
  - $T_k[k - i + 1, i] = 1$,
  - $T_k[k + 1, i] = 1$, except $T_k[k + 1, 1] = 0$,
- for all i in $\{2,\ ..,\ k\}$, $T_k[k - i + 2, i] = 1$.

Table 3. The elements of $\mathcal{H}_k^i$

$n \setminus k$	0	1
0	0	
1	00	01
2	000	001
		011
		0001
3	0000	0011
		0111
		00001
4	00000	00011
		00111
		01111
		000001
		000011
5	000000	000111
		001111
		011111
$\vdots$		

In the other cases, $T_k[i,j] = 0$ and j in $\{1,\ 2,\ ..,\ k\}$.

Remark 3.3. The matrix T_k of our definition establishes the relation between the set $\mathcal{H}_k^1$ and the generating matrix of our code that we will see below. (see also [16]).

Proposition 3.4. *Reminding that $\mathcal{H}_k^1$ is the set of sub-exceeding functions f_i of length $k+1$ such that*

$$f_i = 000 \quad ... \quad 0\underbrace{11...1}_{i-times}\ ,\ \textit{with } i \in \{1, ...k\}\,.$$

Then the product $f_i \times T_k$ gives the word g_i such that

$$\left\{\begin{array}{cc} g_1 = & 0111...111 \\ g_2 = & 1011...111 \\ g_3 = & 1101...111 \\ \vdots & \vdots \\ g_{k-1} = & 1111...101 \\ g_k = & 1111...110 \end{array}\right\}. \tag{6}$$

Notation 3.5. *Now let's denote by G_k the matrix*

$$G_k = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ \vdots \\ g_k \end{pmatrix} = \begin{pmatrix} 0111...111 \\ 1011...111 \\ 1101...111 \\ \vdots \\ 1111...101 \\ 1111...110 \end{pmatrix}. \tag{7}$$

Example 3.6. *For $k = 3$, we have:*

$$G_3 \left\{\begin{array}{cc} g_1 = & 011 \\ g_2 = & 101 \\ g_3 = & 110 \end{array}\right\}, T_3 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \text{ where } \mathcal{H}_3^1 \left\{\begin{array}{cc} f_1 = & 0001 \\ f_2 = & 0011 \\ f_3 = & 0111 \end{array}\right\}$$

For $k = 4$, we have:

$$G_4 \left\{\begin{array}{cc} g_1 = & 0111 \\ g_2 = & 1011 \\ g_3 = & 1101 \\ g_4 = & 1110 \end{array}\right\}, T_4 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \quad \mathcal{H}_4^1 \left\{\begin{array}{cc} f_1 = & 00001 \\ f_2 = & 00011 \\ f_3 = & 00111 \\ f_4 = & 01111 \end{array}\right\}$$

3.2 The linear systematic code $\mathcal{L}_k$

Theorem 3.7. *Let k be a positive integer and let ψ be the linear application from $\mathbb{F}_2^k$ to $\mathbb{F}_2^{2k}$ such that*

$$\begin{array}{rccc} \psi & : & \mathbb{F}_2^k & \longrightarrow & \mathbb{F}_2^{2k} \\ & & m & \longmapsto & \psi(m) = m \times G_{\mathcal{L}_k}, \end{array} \tag{8}$$

where m is the message of k bits such that $m = m_1 m_2 ... m_k$ and $G_{\mathcal{L}_k}$ is the generator matrix such that $G_{\mathcal{L}_k} = \begin{pmatrix} I_k & G_k \end{pmatrix}$, where G_k is the matrix defined in the equation (6).

Thus, the application ψ forms a systematic $[2k, k]$-linear error-correcting code denoted by $\mathcal{L}_k$. The minimum distance of $\mathcal{L}_3$ is 3, and for $k \geq 4$ the minimum distance of $\mathcal{L}_k$ is 4.

Proof. First, since $G_{\mathcal{L}_k} = \begin{pmatrix} I_k & G_k \end{pmatrix}$ is a matrix of k rows and $2k$ columns whose rows are linearly independent vectors, so the application ψ is injective from $\mathbb{F}_2^k$ to $\mathbb{F}_2^{2k}$. Thus, $\psi(\mathbb{F}_2^k)$ is a vector space over $\mathbb{F}_2$ of dimension k. Then ψ forms a systematic linear error-correcting code of dimension k and length $2k$.

Now, let m be the message such as $m = m_1\ m_2 ... m_k$ and note by c its image by the application ψ.

$$c = \psi(m) = m \times G_{\mathcal{L}_k}.$$

Since ψ is a systematic code, a codeword c of length $2k$ can be separated into two vectors c_1 and c_2. That is to say, $c = c_1\ c_2$. Here, the vector c_1 is the original message ($c_1 = m$), and c_2 is the vector (control bits) such that $c_2 = m \times G_k$.

So, for any integer i in $\{1, 2, ..., k\}$, we have

$$c_2[i] = \sum_{j=1, j\neq i}^{k} m_j.$$

So, two cases are possible:

- If the weight of m is even

$$\begin{cases} \text{and that } m_i = 0 & \Rightarrow \quad c_2[i] = \sum_{j=1, j\neq i}^{k} m_j = 0. \\ \text{and if } m_i = 1 & \Rightarrow \quad c_2[i] = \sum_{j=1, j\neq i}^{k} m_j = 1. \end{cases}$$

  In this case, the code word c is: $c = m\ m$.
  (Ex: for $k = 6$, if $m = 011101$, we have $c = 011101\ 011101$)

- If the weight of m is odd

$$\begin{cases} \text{and that } \ m_i = 0 & \Rightarrow \quad c_2[i] = \sum_{j=1, j\neq i}^{k} m_j = 1. \\ \text{and if } \ m_i = 1 & \Rightarrow \quad c_2[i] = \sum_{j=1, j\neq i}^{k} m_j = 0. \end{cases}$$

In this case, the code word c is: $c = m\ \overline{m}$, where $\overline{m}$ is the opposite of m.
(Ex: for $k = 7$, if $m = 0000111$, we have $c = 0000111\ 1111000$)

Now,

1. Take $k = 3$,

$$\begin{array}{rcl} \text{if } w(m) = 1 & \longrightarrow & w(c) = 3, \\ \text{if } w(m) = 2 & \longrightarrow & w(c) = 4, \\ \text{if } w(m) = 3 & \longrightarrow & w(c) = 3. \end{array} \tag{9}$$

Thus, the minimum distance for the code $\mathcal{L}_3$ is 3.

2. for $k \geq 4$,

$$\begin{array}{rcl} \text{if } w(m) = 1 & \longrightarrow & w(c) = k, \\ \text{if } w(m) = 2 & \longrightarrow & w(c) = 4, \\ \text{if } w(m) = 3 & \longrightarrow & w(c) = k, \\ & \vdots & \\ \text{if } w(m) = p \text{ (even)} & \longrightarrow & w(c) = 2p, \\ \text{if } w(m) = q \text{ (odd)} & \longrightarrow & w(c) = k. \end{array} \tag{10}$$

Thus, the minimum distance for the code $\mathcal{L}_k$ is 4.

□

Example 3.8. *For $k = 3$, from the main theorem (3.7), we have*

$$\mathcal{L}_3 = \left\{ \begin{array}{c} 000000 \\ 001110 \\ 010101 \\ 100011 \\ 011011 \\ 101101 \\ 110110 \\ 111000 \end{array} \right\}, \quad G_{\mathcal{L}_3} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

Initial messages:

$$\mathbb{F}_2^3 = \begin{pmatrix} 000 & 010 & 011 & 110 \\ 001 & 100 & 101 & 111 \end{pmatrix}.$$

Example 3.9. *For $k = 4$, from the main theorem (3.7), we have*

$$G_{\mathcal{L}_4} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \tag{11}$$

and

$$\mathcal{L}_4 = \begin{array}{ccc} 00000000 & 00110011 & 01111000 \\ 00011110 & 01010101 & 10110100 \\ 00101101 & 10011001 & 11010010 \\ 01001011 & 01100110 & 11100001 \\ 10000111 & 10101010 & 11111111 \\ & 11001100 & \end{array}$$

3.3 The linear systematic code $\mathcal{L}_k^+$

Theorem 3.10. *Let k be an integer such that $k \geq 4$, and let's take the system $\{e'_1, e'_2, ..., e'_k\}$, where $e'_i = \psi(g_i)$ (see the equation 6). So, we can build a $[3k, k]$-linear systematic code with generator matrix*

$$G_{\mathcal{L}_k^+} = \begin{pmatrix} & e'_1 \\ I_k & \vdots \\ & e'_k \end{pmatrix} \tag{12}$$

which is denoted by $\mathcal{L}_k^+$.

The code $\mathcal{L}_4^+$ has minimum distance 5, and for $k \geq 5$, the code $\mathcal{L}_k^+$ has minimum distance 6. The generating matrix G of this code has the form

$$G_{\mathcal{L}_k^+} = \begin{pmatrix} I_k & G_k & I_k \end{pmatrix}. \tag{13}$$

Proof. Since $\mathcal{L}_k$ is a sub-space over $\mathbb{F}_2$, any linear combination between the code words $e'_1, e'_2, ..., e'_k$ gives a code in $\mathcal{L}_k$ of weight equal to 4. Then, for a message m in $\mathbb{F}_2^k$, the code word c generated by the matrix $G_{\mathcal{L}_k^+}$ (ie $c = m \times G_{\mathcal{L}_k^+}$) has a weight:

1. For $k = 4$,
$$\begin{array}{lll} \text{if } w(m) = 1 & \longrightarrow & w(c') = 5, \\ \text{if } w(m) = 2 & \longrightarrow & w(c') = 6, \\ \text{if } w(m) = 3 & \longrightarrow & w(c') = 7, \\ \text{if } w(m) = 4 & \longrightarrow & w(c') = 12. \end{array} \tag{14}$$
So we have a code 2-corrector $\mathcal{L}_4^+$ with a minimal distance $d = 5$.

2. For $k \geq 5$,
$$\begin{array}{lll} \text{if } w(m) = 1 & \longrightarrow & w(c') \geq 6, \\ \text{if } w(m) = 2 & \longrightarrow & w(c') = 6, \\ \text{if } w(m) = 3 & \longrightarrow & w(c') \geq 7, \\ & \vdots & \\ \text{if } w(m) = k & \longrightarrow & w(c') \geq k + 4. \end{array} \tag{15}$$
So we have a code 2 -corrector $\mathcal{L}_k^+$ with a minimal distance $d = 6$.

□

Example 3.11. *The code $\mathcal{L}_4^+$.*
Now take the four (4) vectors in $\mathcal{L}_4$ which are:

$$\begin{array}{l} e'_1 = 01111000 \\ e'_2 = 10110100 \\ e'_3 = 11010010 \\ e'_4 = 11100001 \end{array} . \tag{16}$$

The code $\mathcal{L}_4^+$ is as follows:

$$G_{\mathcal{L}_4^+} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} . \tag{17}$$

$$\mathcal{L}_4^+ = \begin{array}{c} 0000\ 0000\ 0000 \\ 0001\ 1110\ 0001 \\ 0010\ 1101\ 0010 \\ 0100\ 1011\ 0100 \\ 1000\ 0111\ 1000 \end{array} \quad \begin{array}{c} 0011\ 0011\ 0011 \\ 0101\ 0101\ 0101 \\ 1001\ 1001\ 1001 \\ 0110\ 0110\ 0110 \\ 1010\ 1010\ 1010 \\ 1100\ 1100\ 1100 \end{array} \quad \begin{array}{c} 0111\ 1000\ 0111 \\ 1011\ 0100\ 1011 \\ 1101\ 0010\ 1101 \\ 1110\ 0001\ 1110 \\ 1111\ 1111\ 1111 \end{array}$$

Example 3.12. *The code $\mathcal{L}_5^+$.*
Now take the five (5) vectors in $\mathcal{L}_5$ which are:

$$\begin{array}{l} e'_1 = 01111\ 10000 \\ e'_2 = 10111\ 01000 \\ e'_3 = 11011\ 00100\ . \\ e'_4 = 11101\ 00010 \\ e'_4 = 11110\ 00001 \end{array} \tag{18}$$

The code $\mathcal{L}_5^+$ is thus as follows:

$$G_{\mathcal{L}_5^+} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{19}$$

The code words of $\mathcal{L}_5^+$ are:

$$\begin{array}{ccccc} & 00011\ 00011\ 00011 & 11100\ 00011\ 11100 & \\ & 00101\ 00101\ 00101 & 11010\ 00101\ 11010 & \\ 00000\ 00000\ 00000 & 00110\ 00110\ 00110 & 11001\ 00110\ 11001 & 01111\ 01111\ 01111 \\ 00001\ 11110\ 00001 & 01001\ 01001\ 01001 & 10110\ 01001\ 10110 & 10111\ 10111\ 10111 \\ 00010\ 11101\ 00010 & 01010\ 01010\ 01010 & 10101\ 01010\ 10101 & 11011\ 11011\ 11011 \\ 00100\ 11011\ 00100 & 01100\ 01100\ 01100 & 10011\ 01100\ 10011 & 11101\ 11101\ 11101 \\ 01000\ 10111\ 01000 & 10001\ 10001\ 10001 & 01110\ 10001\ 01110 & 11110\ 11110\ 11110 \\ 10000\ 01111\ 10000 & 10010\ 10010\ 10010 & 01101\ 10010\ 01101 & 11111\ 00000\ 11111 \\ & 10100\ 10100\ 10100 & 01011\ 10100\ 01011 & \\ & 11000\ 11000\ 11000 & 00111\ 11000\ 00111 & \end{array} \tag{20}$$

4 Decoding for the error correcting codes $\mathcal{L}_k$ and $\mathcal{L}_k^+$

After considering the parameters necessary for the study of these codes, we present here the appropriate decoding algorithms.

4.1 The dual codes of $\mathcal{L}_k$ and $\mathcal{L}_k^+$

Theorem 4.1. *(See [9], [12], [17])*
If C is an $[n,k]$ code over $\mathbb{F}_2$, then the dual code $C^\perp$ is given by all words $u \in \mathbb{F}_2^n$ such that $< u\ ,\ c >= 0$ for each $c \in C$, where $<\ ,\ >$ denotes the ordinary inner product. The dual code $C^\perp$ is an $[n, n-k]$ code. If $G = (I_k \mid M)$ is a generator matrix for C, then $H = \left(M^T \mid I_{n-k}\right)$ is the generator matrix for $C^\perp$.

Example 4.2. *For the code $\mathcal{L}_4$ and $\mathcal{L}_4^+$ the generator matrix is respectively*

$$G_{\mathcal{L}_4} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

and

$$G_{\mathcal{L}_4^+} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

So the dual code $\mathcal{L}_4^\perp$ and $(\mathcal{L}_4^+)^\perp$ have respectively his generator matrix:

$$H_{\mathcal{L}_4^\perp} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

and

$$H_{(\mathcal{L}_4^+)^\perp} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{21}$$

As the columns of $H_{\mathcal{L}_4^\perp}$ (or $H_{(\mathcal{L}_4^+)^\perp}$) are pairwise distinct, so for a codeword c that contains exactly one error, the decoding will be easy by looking at the $H \times {}^t c$ syndrome.

Definition 4.3. Let c be an element of linear code C such that $c = m_1\ m_2\ ...\ m_n$, where $m_i \in \mathbb{F}_2$ for all i. We define the monomial X^c of $\mathbb{F}_2[X_1X_2...X_n]$ by

$$X^c = X_1^{m_1} X_2^{m_2} ... X_n^{m_n}. \tag{22}$$

Example 4.4. *Take $c = 101101$ which is a codeword of $\mathcal{L}_3$. In $\mathbb{F}_2[X_1X_2...X_6]$, the monomial X^c was*

$$X^c = X_1^1 X_2^0 X_3^1 X_4^1 X_5^0 X_6^1 = X_1X_3X_4X_6.$$

Theorem 4.5. *(see [4])*
Let C be an $[n,k]$-linear systematic code over $\mathbb{F}_2$. Define by I_C the binomial ideal of $\mathbb{F}_2[X_1X_2...X_n]$ associated with C such that

$$I_C = \langle X^c - X^{c'} \mid c - c' \in C \rangle + \langle X_i^2 - 1 \mid 1 \leq i \leq n \rangle. \tag{23}$$

Theorem 4.6 (Groebner basis of the Binomial ideal I_C)**.** *(See [12])*
Take the lexicographic order on $\mathbb{F}_2[X_1X_2...X_n]$, i.e. $X_1 \succ ... \succ X_n$. An $[n,k]$ linear systematic code C of generator matrix $(I_k \mid M)$ has the reduced Groebner basis

$$\mathcal{B} = \{X_i - X^{m_i} \mid 1 \leq i \leq k\} \cup \left\{X_i^2 - 1 \mid k+1 \leq i \leq n\right\}. \tag{24}$$

Here m_i is the i^{th} line of the matrix M.

Example 4.7. *For the code* $(\mathcal{L}_4^+)$*, the corresponding binomial ideal of this code in* $\mathbb{F}_2[X_1X_2...X_n]$ *has the reduced Groebner basis given by the elements*

$$\begin{array}{ll} b_1 = X_1 - X_6X_7X_8X_9 & b_5 = X_5^2 - 1 \\ b_2 = X_2 - X_5X_7X_8X_{10} & b_6 = X_6^2 - 1 \\ b_3 = X_3 - X_5X_6X_8X_{11} & b_7 = X_7^2 - 1 \\ b_4 = X_4 - X_5X_6X_7X_{12} & b_8 = X_8^2 - 1 \\ b_9 = X_9^2 - 1 & b_{10} = X_{10}^2 - 1 \\ b_{11} = X_{11}^2 - 1 & b_{12} = X_{12}^2 - 1 \end{array}, \tag{25}$$

where

$$G_{\mathcal{L}_4^+} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{26}$$

4.2 Error-correction of the code $\mathcal{L}_k$

1. The ideal case is that no error was produced during transmission. We can use two methods to detect the presence of errors:
   - We make the product of the control matrix H with the received code and we have to find a null vector, which means that there was no error during the transmission.
   - Now the second method: as our code $\mathcal{L}_k$ is a systematic code, the received code word can be split into two, i.e. $c = m_1\ m_2$, where m_1 is the message sent and m_2 is the control code. So, if the weight of m_1 is even and $m_1 = m_2$ or if the weight of m_1 is odd and $m_2 = \overline{m_1}$, in both cases the code has no error during the transmission. Otherwise there are errors.
2. The other case is that errors occur during transmission.
   Suppose that one error was produced. So, we find out here how to fix it. The only error must be in m_1 or m_2. Moreover, if the real message m sent is of even (odd) weight, the word $m_1 + m_2$ is of weight 1 (resp $(k-1)$). As a result, the decoding is as follows:

- If the weight of $m_1 + m_2$ is 1, it remains to find the only bit that distinguishes m_1 from m_2 and fix it for the weight of m_1 to be even.
- If the weight of $m_1 + m_2$ is $k-1$, it remains to find the only bit for that $m_2 = \overline{m_1}$ and fix it for the weight of m_1 to be odd.

Algorithm of Decoding for the code $\mathcal{L}_k$

Input r (*received word*)

Output c (*corrected word*)

Begin

Determine m_1 and m_2 such that $r = m_1 m_2$;
Calculate $w_1 = w(m_1)$, $w_2 = w(m_2)$
and $w_{1,2} = w(m_1 + m_2)$;

- If $w_{1,2} = 0$ or $w_{1,2} = k$, so $c = r$;
- If $w_{1,2} = 1$
  - and if w_1 is even, so $c = m_1\ m_1$;
  - and if w_1 is odd, so $c = m_2\ m_2$;
- If $w_{1,2} = k - 1$
  - and if w_1 is odd, so $c = m_1\ \overline{m_1}$;
  - and if w_1 is even, so $c = \overline{m_2}\ m_2$;
- Else print("The message contains more than one error, we can not correct them")

End

By simple calculation, we find that the complexity of this algorithm is linear, i.e. $\mathcal{O}(n)$.

Remark 4.8. For $k = 4$, the parameters of the code $\mathcal{L}_4$ coincide with those of the extended Hamming code $H(8, 4, 4)$. But by comparing the decoding algorithm presented in [14] on page 88 to 92 (decoding by the butterfly operator) with our algorithm, the complexity of our decoding is interesting.

Remark 4.9. The code Hadamard$[4, 2, 2]$ have also the same parameters as our code $\mathcal{L}_2$.

4.3 Error correction for the code $\mathcal{L}_k^+$

We try to give here the correction steps for a codeword that contains at most 2 errors.
An immediate consequence of the study of the reduced Groebner basis of the Binomial ideal I_C is a decoding algorithm for linear codes. This algorithm was given in slightly different form in [4].

Theorem 4.10. *(See [12])*
Let C be an [n,k] code over $\mathbb{F}_2$, and let $\mathcal{B}$ be the reduced Groebner basis for C given in (24)*. Suppose the code C is t-error-correcting. The following algorithm gives a decoder D for the code C. Given a received word $c \in \mathbb{F}_2^n$, if the word given by $rem(X^c - 1,\ \mathcal{B})$ has at most t nonzero entries, then form $D(c) = (X^c - 1) - rem(X^c - 1,\ \mathcal{B})$. This gives the codeword that is closest to the received word.*

Remark 4.11. In other ways, for the linear systematic code $\mathcal{L}_k^+$, we can also use the parity check matrix $H_{\mathcal{L}_k^+}$ for the decoding. By the form of this parity check matrix, we have:

- all the columns of $H_{\mathcal{L}_k^+}$ (see (21)) are different from each other,
- all additions of two columns of $H_{\mathcal{L}_k^+}$ are also pairwise distinct.

Then, for a received word c which contains at most two errors,

- if one error was presented at the i^{th} position of c, thus $H \times {}^t c = h_i$.
- If c contains two errors at the i^{th} and j^{th} position $(i < j)$, thus $H \times {}^t c = h_i + h_j$.

The calculation of $H_{\mathcal{L}_k^+} \times {}^t c$ specifies the positions of errors. So, we find that the number of operations for the decoding of the code $\mathcal{L}_k^+$ denoted N_{op} is $2k(6k + 1)$ i.e.

$$N_{op}(\mathcal{L}_k^+) = \frac{2n(2n+1)}{3}, \text{ where } n = 3k.$$

Then, we find that the complexity of this algorithm is quadratic, i.e. $\mathcal{O}(n^2)$.

4.3.1 Comparative analysis between $\mathcal{L}_k^+$ and the code of Hamming

Table 4 below presents a comparative study between the Hamming code and $\mathcal{L}_k^+$ for some cases where the two codes are of the same length.

Table 4. Comparative study between Hamming code and $\mathcal{L}_k^+$

	HAMMING CODE	CODE BUILT FROM SUB-EXCEEDING FONCTION
Form	It is a linear code of the form $[2^r-1, 2^r-r-1, 3]$. For $r=4$, we have $[15, 11, 3]-$ linear code. For $r=6$, we have $[63, 57, 3]-$ linear code.	It is a linear code of the form $[3k, k, 6]$. For $k=5$, we have $[15, 5, 6]-$ linear code. For $k=21$, we have $[63, 21, 6]-$ linear code.
Parameters	Minimum distance: $d=3$ Correction capacity: $e_c=1$ For the $[15, 11, 3]-$ code Correction rate is $C_r=1/15$ For the $[63, 57, 3]-$ code Correction rate is $C_r=1/63$	Minimum distance: $d=6$ Correction capacity: $e_c=2$ For the $[15, 5, 6]-$ code Correction rate is $C_r=2/15$ For the $[63, 21, 6]-$ code Correction rate is $C_r=2/63$

These codes have the same length but different dimensions. However, the code $\mathcal{L}_k^+$ has 2 bits for the correction capability and the error detection capability was 5 comparing with the Hamming code which can correct one error and detect only 2 errors.

References

[1] Pierre Abbrugiati, *Introduction to error correcting codes*, 23 January 2006, 36 p. (in French)

[2] W. Adams, P. Loustaunau, *An Introduction to Groebner Bases* (Graduate Studies in Mathematics, vol. 3), Providence, RI, USA: American Mathematical Society, 1994, 289 p. ISBN:0-8218-3804-0.

[3] T. Becker, V. Weispfenning, *Groebner Bases. A Computational Approach to Commutative Algebra* (Graduate Texts in Mathematics, vol. 141), New York: Springer, 1998, 576 p. ISBN-10: 0387979719. ISBN-13: 978-0387979717.

[4] M. Borges-Quintana, M.A. Borges-Trenard, P. Fitzpatrick, E. Martinez-Moro, "Groebner bases and combinatorics for binary codes," *Applicable Algebra in Engineering, Communication and Computing*, vol. 19, no. 5, pp. 393–411, 2008.

[5] M. J. E. Golay, "Notes on digital coding," in *Proc. IRE*, 1949, vol. 37, pp. 657.

[6] *Combinatorial Analysis* (Proceedings of Symposia in Applied Mathematics, vol. 10), R. Bellman and M. Hall Jr., Eds. Providence, R.I., USA: American Mathematical Society, 1960, 311p. Electronic ISBN: 978-0-8218-9225-1.

[7] D. Knuth, *The Art of Computer Programming – Sorting and Searching*, 2nd ed., vol. 3, Redwood City, CA, USA: Addison Wesley Longman Publishing Co. Inc., 1998, xiv+780p. ISBN: 0-201-89685-0.

[8] D.H. Lehmer, "Teaching combinatorial tricks to a computer," in *Combinatorial Analisis* (Proc. Sympos. Appl. Math., vol. 10), Providence, R.I., USA: Amer. Math. Soc, 1960, pp. 179–193.

[9] F.J. MacWilliams and N.J.A. Sloane, *The theory of error correcting codes* (North-Holland mathematical library, vol. 16), Amsterdam-New York-Oxford, Netherlands: North-Holland Pub. Co., 1977, xx+762 p. ISBN: 0444850090 and 0444850104.

[10] Roberto Mantaci, "On the distribution of anti-excesses in the symmetric group and its subgroups," *Theoret. Comp. Science*, vol. 117, no. 1–2, pp. 243–253, 1993. (in French).

[11] Roberto Mantaci and Fanja Rakotondrajao, "A permutation that knows what Eulerian means," *Discrete Mathematics and Theortical Computer Science*, vol. 4, pp. 101–108, May 2001.

[12] Mehwish Saleemi and Karl-Heinz Zimmermann, "Groebner Bases For Linear Codes," *International Journal of Pure and Applied Mathematics*, vol. 62, no. 4, pp. 481–491, 2010.

[13] Melvyn el Kamel-Meurigne (Under the direction of Benoit Fabrèges), *Error correcting code*, 23 January 2006, 31 p. (in French).

[14] Senad Mohamed-Mahmoud, "Contribution to soft decision decoding of bulk correcting codes," Ph.D. dissertation, January 2016, HAL Id: tel-01356210. (in French).

[15] L. Rabefihavanana, "Encoding of Partition Set Using Sub-exceeding Function," *International Journal of Contemporary Mathematical Sciences*, vol. 13, no. 2, pp. 63–78, 2018.

[16] Luc Rabefihavanana, "Parts of a set and sub-exceeding function: Coding and Decoding," *Bulletin Of Mathematics And Statistics Research*, vol. 5., no. 3, pp, 38–53, 2017 (July-Sept) Ky Publications.

[17] J. H. van Lint, *Introduction to Coding Theory* (Graduate Texts in Mathematics, vol. 86), Berlin: Springer-Verlag Berlin Heidelberg, 1999, xii+186 p.

L. Rabefihavanana, H. Andriatahiny,
T. Rabeherimanana

Received November 16, 2018
Revised January 21, 2019

Luc Rabefihavanana
Department of Mathematics and Computer Science
Faculty of Sciences, University of Antananarivo, Madagascar
Phone: (+261) 34 74 877 40
E–mail: lucrabefihavanana@yahoo.fr

Harinaivo Andriatahiny
Department of Mathematics and Computer Science
Faculty of Sciences, University of Antananarivo, Madagascar
E–mail: aharinaivo@yahoo.fr

Toussaint Joseph Rabeherimanana
Department of Mathematics and Computer Science
Faculty of Sciences, University of Antananarivo, Madagascar
E–mail: rabeherimanana.toussaint@yahoo.fr

Computer Science Journal of Moldova, vol.**27**, no.1(79), 2019

Post-quantum public key-agreement scheme based on a new form of the hidden logarithm problem*

D.N. Moldovyan

Abstract

A new form of the hidden discrete logarithm problem, proposed as primitive of the post-quantum public-key cryptoschemes, is defined over the 6-dimensional finite non-commutative associative algebra with a large set of the left-sided global units. The considered computationally difficult problem uses the mutual commutativity of the exponentiation operation and homomorphism mapping defined relatively a fixed unit element of the algebra. The related properties of the introduced algebra are described. Novel public key-agreement and zero-knowledge protocols based on the hidden logarithm problem are introduced as post-quantum cryptoschemes.

Keywords: finite non-commutative algebra, associative algebra, computationally difficult problem, homomorphism, key agreement scheme, zero-knowledge protocol, post-quantum cryptoscheme

MSC 2010: 94A60, 16Z05, 14G50, 11T71, 16S50.

1 Introduction

Currently the most widely used in practice public key cryptographic algorithms and protocols are based on computational difficulty of the factoring problem (FP) and discrete logarithm problem (DLP) [1]. The FP and DLP are universal cryptographic primitives that allow

*The work is supported by the Russian Foundation for Basic Research grant # 18-07-00932-a.

designing on their base cryptoschemes of different types: public encryption and digital signature algorithms, public key-agreement and zero-knowledge authentication protocols, blind signature and multisignature schemes.

Both the FP and the DLP can be solved in polynomial time on a quantum computer [2], [3], therefore due to the latest progress in the quantum computation technology the cryptographic community has faced the challenge of developing the post-quantum public-key cryptoschemes, i. e., the cryptoschemes that resist the attacks with using both the ordinary computers and the quantum ones [4], [5]. Developers of the post-quantum public-key cryptoschemes look for suitable computational difficult problems different from the FP and DLP, which have superpolynomial computational difficulty when solving them on a quantum computer.

One of the latter computationally difficult problems, called conjugacy search problem, had been defined over braid groups representing a particular type of non-commutative groups [6], [7]. Using computations in the braid groups a number of different public-key cryptoschemes have been developed [8], [9]. The articles [10], [11] highlight some fundamental difficulties in using the conjugacy problem for developing cryptoschemes with superpolynomial security.

A promising approach to design the post-quantum cryptoschemes relates to combining the DLP with the problem of finding the conjugating element, which leads to the so-called hidden DLP (HDLP), i. e., to the DLP in a cyclic group hidden in the finite algebra of quaternions defined over the ground field $GF(p)$ [12], [13]. The computational complexity of the HDLP is superpolynomial in solving it on conventional computers. However, in [14] the polynomial reducibility of the HDLP in finite algebra of quaternions to the DLP in the field $GF(p^2)$ was shown. Therefore, using the form of the HDLP described in [12], [13] and the finite algebra of quaternions as its algebraic support can not be considered as a direct way to developing the post-quantum public-key algorithms and protocols. The search for new algebraic supports and new forms of the HDLP was noted in [14]–[16] as one of the conditions for the development of post-quantum cryptoschemes based on the

HDLP. In papers [15],[16] different types of the finite non-commutative associative algebras (FNAAs) are proposed as algebraic supports of the HDLP defined in new forms. New forms of the HDLP proposed in [16] are used as the base primitive for two different post-quantum digital signature schemes, however that forms of the HDLP do not suite for designing the public key-agreement schemes.

In present paper it is introduced a new form of the HDLP defined over the 6-dimensional FNAA containing a large set of the global left-sided units. The HDLP is defined relatively some specified left-sided unit and properties of the used FNAA connected with its unit elements are investigated. The public key-agreement scheme and zero-knowledge protocol based on the proposed form of the HDLP are introduced as candidates of the public-key post-quantum cryptoschemes.

2 The used 6-dimensional FNAA

Suppose the m-dimensional vector space is defined over the finite ground field $GF(p)$. Defining additionally the multiplication operation that is distributive relatively the addition operation one gets the m-dimensional finite algebra. If the multiplication operation is associative and non-commutative, then the algebra is called FNAA. To define the multiplication operation in the vector space one can use the representation of some vector $A = (a_0, a_1, \dots a_{m-1})$ in the form of the following summ of the single component vectors $a_i\mathbf{e}_i$: $A = \sum_{i=0}^{m-1} a_i\mathbf{e}_i$, where $\mathbf{e}_0 = (1, 0, 0 \dots, 0)$, $\mathbf{e}_1 = (0, 1, 0 \dots, 0)$, ... $\mathbf{e}_{m-1} = (0, 0 \dots, 0, 1)$ are formal basis vectors.

The multiplication operation $\circ$ of two m-dimensional vectors A and $B = \sum_{j=0}^{m-1} b_j\mathbf{e}_j$ is defined as follows:

$$A \circ B = \left(\sum_{i=0}^{m-1} a_i\mathbf{e}_i\right) \circ \left(\sum_{j=0}^{m-1} b_j\mathbf{e}_j\right) = \sum_{j=0}^{m-1}\sum_{i=0}^{m-1} a_i b_j \left(\mathbf{e}_i \circ \mathbf{e}_j\right), \qquad (1)$$

where every product of two babis vectors $\mathbf{e}_i \circ \mathbf{e}_j$ is to be replaced by some single-component vector $\lambda\mathbf{e}_k$ that is taken from the so called basis

vector multiplication table (BVMT), like Table 1. It is assumed that the intersection of the ith row and the jth column defines the cell indicating the value $\lambda\mathbf{e}_k = \mathbf{e}_i \circ \mathbf{e}_j$. If the structural coefficient $\lambda = 1$, then the mentioned single-component vector is written as $\mathbf{e}_k$.

To define a FNAA one should compose respective BVMT that defines non-commutative asssociative multiplication operation. A unified method for defining the FNAAs of arbitrary dimensions $m > 1$ is proposed in [17]. Investigation of the general properties of that FNAAs had shown that their characteristic property is the existence of large set of the global single-sided units. The FNAAs possessing such property are very interesting for using them as algebraic support of the public key-agreement schemes, in this case a new form of the HDLP is to be proposed though. The last is because the known form of the HDLP used in such type of cryptoschemes exploits existance of the global two-sided unit.

Since the FNAAs described in [17] allows trivial reduction of the HDLP to the DLP in the finite field $GF(p)$, in this paper a new 6-dimensional FNAA containing p^4 global left-sided units is introduced with the BVMT composed as follows. Initially a preliminary BVMT have been built using the following formula:

$$\mathbf{e}_i \circ \mathbf{e}_j = \mathbf{e}_{3i+j}, \tag{2}$$

where $i, j = 0, 1, \ldots, 5$ and computation of the value $3i+j$ is performed modulo 6.

Proposition 1. The formula (2) together with the formula (1) defines non-commutative associative multiplication operation of the 6-dimensional vectors.

Proof. Using the formula (1) one can get the following formula describing the product of the vectors A, B, and $C = \sum_{k=0}^{5} c_k\mathbf{e}_k$:

$$(A \circ B) \circ C = \sum_{i=0}^{5}\sum_{j=0}^{5}\sum_{k=0}^{5} a_i b_j c_k \left(\mathbf{e}_i \circ \mathbf{e}_j\right) \circ \mathbf{e}_k;$$

$$A \circ (B \circ C) = \sum_{i=0}^{5}\sum_{j=0}^{5}\sum_{k=0}^{5} a_i b_j c_k \mathbf{e}_i \circ \left(\mathbf{e}_j \circ \mathbf{e}_k\right).$$

This formula shows the multiplication operation is associative, since the formula (2) defines associative multiplication of every possible triple of the basis vectors. Indeed, for arbitrary possible three values i, j, and k we have the following

$$\left.\begin{array}{l}(\mathbf{e}_i \circ \mathbf{e}_j) \circ \mathbf{e}_k = \mathbf{e}_{3i+j} \circ \mathbf{e}_k = \mathbf{e}_{9i+3j+k} = \mathbf{e}_{3i+3j+k} \\ \mathbf{e}_i \circ (\mathbf{e}_j \circ \mathbf{e}_k) = \mathbf{e}_i \circ \mathbf{e}_{3j+k} = \mathbf{e}_{3i+3j+k}\end{array}\right\} \Rightarrow$$

$$\Rightarrow (\mathbf{e}_i \circ \mathbf{e}_j) \circ \mathbf{e}_k = \mathbf{e}_i \circ (\mathbf{e}_j \circ \mathbf{e}_k)\,.$$

Thus, the formula (2) describes some preliminary BVMT defining the associative multiplication of the basis vectors. Then we have found experimentally the distributions of the structural coefficients $\lambda \neq 1$ and $\epsilon \neq 1$, which retain the property of the associativity of the multiplication operation. The resultant BVMT is shown as Table 1.

Table 1. The BVMT defining the 6-dimensional FNAA with p^4 different global left-sided units.

$\circ$	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$
$\mathbf{e}_0$	$\lambda\mathbf{e}_0$	$\lambda\mathbf{e}_1$	$\lambda\mathbf{e}_2$	$\lambda\mathbf{e}_3$	$\lambda\mathbf{e}_4$	$\lambda\mathbf{e}_5$
$\mathbf{e}_1$	$\epsilon\mathbf{e}_3$	$\epsilon\mathbf{e}_4$	$\epsilon\mathbf{e}_5$	$\epsilon\mathbf{e}_0$	$\epsilon\mathbf{e}_1$	$\epsilon\mathbf{e}_2$
$\mathbf{e}_2$	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$
$\mathbf{e}_3$	$\lambda\mathbf{e}_3$	$\lambda\mathbf{e}_4$	$\lambda\mathbf{e}_5$	$\lambda\mathbf{e}_0$	$\lambda\mathbf{e}_1$	$\lambda\mathbf{e}_2$
$\mathbf{e}_4$	$\epsilon\mathbf{e}_0$	$\epsilon\mathbf{e}_1$	$\epsilon\mathbf{e}_2$	$\epsilon\mathbf{e}_3$	$\epsilon\mathbf{e}_4$	$\epsilon\mathbf{e}_5$
$\mathbf{e}_5$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$

2.1 Some properties related to defining the HDLP

The left-sided units of the algebra introduced in Section 2 can be described as solutions of the following vector equation

$$X \circ A = A. \tag{3}$$

Using Table 1 one can represent (3) in the form of the following system of six linear equations with coordinates of the left operand $x_0, x_1, \ldots, x_5$

as the unknown values:

$$\begin{cases} \lambda x_0 a_0 + \epsilon x_1 a_3 + x_2 a_0 + \lambda x_3 a_3 + \epsilon x_4 a_0 + x_5 a_3 = a_0; \\ \lambda x_0 a_1 + \epsilon x_1 a_4 + x_2 a_1 + \lambda x_3 a_4 + \epsilon x_4 a_1 + x_5 a_4 = a_1; \\ \lambda x_0 a_2 + \epsilon x_1 a_5 + x_2 a_2 + \lambda x_3 a_5 + \epsilon x_4 a_2 + x_5 a_5 = a_2; \\ \lambda x_0 a_3 + \epsilon x_1 a_0 + x_2 a_3 + \lambda x_3 a_0 + \epsilon x_4 a_3 + x_5 a_0 = a_3; \\ \lambda x_0 a_4 + \epsilon x_1 a_1 + x_2 a_4 + \lambda x_3 a_1 + \epsilon x_4 a_4 + x_5 a_1 = a_4; \\ \lambda x_0 a_5 + \epsilon x_1 a_2 + x_2 a_5 + \lambda x_3 a_2 + \epsilon x_4 a_5 + x_5 a_2 = a_5. \end{cases} \tag{4}$$

The system (4) can be rewritten in the following form

$$\begin{cases} (\lambda x_0 + x_2 + \epsilon x_4)\, a_0 + (\epsilon x_1 + \lambda x_3 + x_5)\, a_3 = a_0; \\ (\epsilon x_1 + \lambda x_3 + x_5)\, a_0 + (\lambda x_0 + x_2 + \epsilon x_4)\, a_3 = a_3; \\ (\lambda x_0 + x_2 + \epsilon x_4)\, a_1 + (\epsilon x_1 + \lambda x_3 + x_5)\, a_4 = a_1; \\ (\epsilon x_1 + \lambda x_3 + x_5)\, a_1 + (\lambda x_0 + x_2 + \epsilon x_4)\, a_4 = a_4; \\ (\lambda x_0 + x_2 + \epsilon x_4)\, a_2 + (\epsilon x_1 + \lambda x_3 + x_5)\, a_5 = a_2; \\ (\epsilon x_1 + \lambda x_3 + x_5)\, a_2 + (\lambda x_0 + x_2 + \epsilon x_4)\, a_5 = a_5. \end{cases} \tag{5}$$

Performing the variable substitution $u_0 = \lambda x_0 + x_2 + \epsilon x_4$ and $u_1 = \epsilon x_1 + \lambda x_3 + x_5$ in the system (5) one gets the following three systems of two linear equations

$$\begin{cases} a_0 u_0 + a_3 u_1 = a_0; \\ a_3 u_0 + a_0 u_1 = a_3; \end{cases} \tag{6}$$

$$\begin{cases} a_1 u_0 + a_4 u_1 = a_1; \\ a_4 u_0 + a_1 u_1 = a_4; \end{cases} \tag{7}$$

$$\begin{cases} a_2 u_0 + a_5 u_1 = a_2; \\ a_5 u_0 + a_2 u_1 = a_5. \end{cases} \tag{8}$$

It is easy to see that for arbitrary vector A each of the equations of every of the systems (6), (7), and (8) holds true for the values of the unknowns

$u_0 = 1$ and $u_1 = 0$. This means that every vector X coordinates of which satisfy the conditions

$$u_0 = \lambda x_0 + x_2 + \epsilon x_4 = 1 \quad \text{and} \quad u_1 = \epsilon x_1 + \lambda x_3 + x_5 = 0 \qquad (9)$$

acts as the global left-sided units of the considered FNAA, i. e., the left-sided units acting on all elements of the algebra. Thus, we have the following formula describing the set of all p^4 global left-sided unit elements $L = (l_0, l_1, l_2, l_3, l_4, l_5)$:

$$L = (x_0, x_1, 1 - \lambda x_0 - \epsilon x_4, x_3, x_4, -\epsilon x_1 - \lambda x_3)\,, \qquad (10)$$

where $x_0, x_1, x_3, x_4 = 0, 1, 2, \ldots, p-1$.

Considering the systems (6), (7), and (8) one can state existence of two particular sets of the vectors $A = (a_0, a_1, a_2, a_3, a_4, a_5)$. The first particular set includes the vectors coordinates of which satisfy simultaneously the following three conditions $a_0 = a_3$, $a_1 = a_4$, and $a_2 = a_5$. The second particular set includes vectors satisfying simultaneously the conditions $a_0 = -a_3$, $a_1 = -a_4$, and $a_2 = -a_5$. To each of these particular sets there corresponds a large set of local left-sided units, i. e., the left-sided units acting only on vectors contained in the particular sets.

The local units acting on the vectors from the first set is described as follows

$$L' = (x_0, x_1, x_2, x_3, x_4, 1 - \lambda x_0 - x_2 - \epsilon x_4 - \epsilon x_1 - \lambda x_3)\,,$$

where $x_0, x_1, x_2, x_3, x_4 = 0, 1, 2, \ldots, p-1$.

The local units acting on the vectors from the second set is described as follows

$$L'' = (x_0, x_1, x_2, x_3, x_4, \lambda x_0 + x_2 + \epsilon x_4 - \epsilon x_1 - \lambda x_3 - 1)\,,$$

where $x_0, x_1, x_2, x_3, x_4 = 0, 1, 2, \ldots, p-1$.

For every vector A that is not contained in the first nor in the second set, the systems (6), (7), and (8) are satisfied simultaneously only for the case $u_0 = 1$ and $u_1 = 0$. Therefore, only the global left-sided units

act on every vector A that is not contained in the indicated particular sets.

Computing the right-sided units for some fixed vector A is connected with finding solutions of the vector equation

$$A \circ X = A. \tag{11}$$

Using Table 1 one can represent (11) in the form of the following system of six linear equations with coordinates of the right operand $x_0, x_1, \ldots, x_5$ as the unknown values:

$$\begin{cases} \lambda a_0 x_0 + \epsilon a_1 x_3 + a_2 x_0 + \lambda a_3 x_3 + \epsilon a_4 x_0 + a_5 x_3 = a_0; \\ \lambda a_0 x_1 + \epsilon a_1 x_4 + a_2 x_1 + \lambda a_3 x_4 + \epsilon a_4 x_1 + a_5 x_4 = a_1; \\ \lambda a_0 x_2 + \epsilon a_1 x_5 + a_2 x_2 + \lambda a_3 x_5 + \epsilon a_4 x_2 + a_5 x_5 = a_2; \\ \lambda a_0 x_3 + \epsilon a_1 x_0 + a_2 x_3 + \lambda a_3 x_0 + \epsilon a_4 x_3 + a_5 x_0 = a_3; \\ \lambda a_0 x_4 + \epsilon a_1 x_1 + a_2 x_4 + \lambda a_3 x_1 + \epsilon a_4 x_4 + a_5 x_1 = a_4; \\ \lambda a_0 x_5 + \epsilon a_1 x_2 + a_2 x_5 + \lambda a_3 x_2 + \epsilon a_4 x_5 + a_5 x_2 = a_5. \end{cases} \tag{12}$$

The system (12) can be represented in the form of the following three independent systems of two linear equations

$$\begin{cases} (\lambda a_0 + a_2 + \epsilon a_4)\, x_0 + (\epsilon a_1 + \lambda a_3 + a_5)\, x_3 = a_0; \\ (\epsilon a_1 + \lambda a_3 + a_5)\, x_0 + (\lambda a_0 + a_2 + \epsilon a_4)\, x_3 = a_3; \end{cases} \tag{13}$$

$$\begin{cases} (\lambda a_0 + a_2 + \epsilon a_4)\, x_1 + (\epsilon a_1 + \lambda a_3 + a_5)\, x_4 = a_1; \\ (\epsilon a_1 + \lambda a_3 + a_5)\, x_1 + (\lambda a_0 + a_2 + \epsilon a_4)\, x_4 = a_4; \end{cases} \tag{14}$$

$$\begin{cases} (\lambda a_0 + a_2 + \epsilon a_4)\, x_2 + (\epsilon a_1 + \lambda a_3 + a_5)\, x_5 = a_2; \\ (\epsilon a_1 + \lambda a_3 + a_5)\, x_2 + (\lambda a_0 + a_2 + \epsilon a_4)\, x_5 = a_5. \end{cases} \tag{15}$$

The same main determinant Δ corresponds to each of the systems (13), (14), and (15):

$$\Delta_A = \alpha^2 - \beta^2, \tag{16}$$

where $\alpha = \lambda a_0 + a_2 + \epsilon a_4$ and $\beta = \epsilon a_1 + \lambda a_3 + a_5$. If $\Delta \neq 0$, then every of the systems (13), (14), and (15) has a unique solution, i. e., the vector

equation (11) also has a unique solution as the single right-sided unit R related to the vector A. It is easy to write the following formulas describing the vector $R_A = (r_0, r_1, r_2, r_3, r_4, r_5)$:

$$\begin{array}{lll} r_0 = \dfrac{a_0\alpha - a_3\beta}{\Delta_A}; & r_1 = \dfrac{a_1\alpha - a_4\beta}{\Delta_A}; & r_2 = \dfrac{a_2\alpha - a_5\beta}{\Delta_A}; \\ r_3 = \dfrac{a_3\alpha - a_0\beta}{\Delta_A}; & r_4 = \dfrac{a_4\alpha - a_1\beta}{\Delta_A}; & r_5 = \dfrac{a_5\alpha - a_2\beta}{\Delta_A}. \end{array} \tag{17}$$

Proposition 2. Suppose the vector A is such that $\Delta_A \neq 0$. Then the local right-sided unit R_A relating to A is contained in the set of the global left-sided units, i. e., there exists the single local two-sided unit E_A relating to the vector A that is equal to R_A.

Proof. Let us consider the formula (10) describing the set of the global left-sided units. Suppose $x_0 = r_0$, $x_1 = r_1$, $x_3 = r_3$, $x_4 = r_4$. Taking into account the formulas (17) computation of the coordinate x_2 gives the following

$$x_2 = 1 - \lambda r_0 - \epsilon r_4 = \frac{1}{\Delta_A}\left(\alpha^2 - \beta^2 - \lambda a_0\alpha + \lambda a_3\beta - \epsilon a_4\alpha + \epsilon a_1\beta\right) = \\ = \frac{1}{\Delta_A}\left(a_2\alpha - a_5\beta\right) = r_2.$$

Computation of the coordinate x_5 gives the following

$$x_5 = 1 - \epsilon r_1 - \lambda r_3 = \frac{1}{\Delta_A}\left(-\epsilon a_1\alpha + \epsilon a_4\beta - \lambda a_3\alpha + \lambda a_0\beta\right) = \\ = \frac{1}{\Delta_A}\left(a_5\left(\lambda a_0 + \epsilon a_4\right) - a_2\left(\epsilon a_1 + \lambda a_3\right)\right) = r_5.$$

Thus, the vector R_A is contained in the set of the global right-sided units (10). Proposition 2 is proven.

Proposition 3. Suppose the vector A is such that $\Delta_A \neq 0$. Then the local right-sided unit R_A relating to A relates also to the vector A^i for arbitrary natural value i.

Proof.

$$\left\{A \circ R_A = R_A \circ A = A\right\} \Rightarrow \\ \left\{A^i \circ R_A = A^{i-1} \circ A \circ R_A = A^i;\ R_A \circ A^i = R_A \circ A \circ A^{i-1} = A^i\right\}.$$

Proposition 4. Suppose the vector A is such that $\Delta_A \neq 0$. Then the sequence $A, A^2, \ldots, A^i, \ldots$ is periodic and for some positive integer ω we have $A^\omega = R_A$.

Proof. Suppose the sequence $A, A^2, \ldots, A^i, \ldots$ contains the zero vector $O = (0, 0, 0, 0, 0, 0)$. Then for some natural number j (for example, $j = 2$) we have $A^{j-1} \neq O$ and $A^j = O$, i. e., $A \circ A^{j-1} = O$. Since $\Delta_A \neq 0$ and $X = O$ satisfies the equation $A \circ X = O$, the last equation has a unique solution $X = O$. Therefore, $A^{j-1} = O$. The obtained contradiction proves that all values in the considered sequence are different from O.

The last fact and the finiteness of the considered algebra shows that for some natural numbers i and $t > i$ we have

$$\left\{A^i = A^t = A^{t-i} \circ A^i = A^i \circ A^{t-i}\right\} \Rightarrow E_{A^i} = A^{t-i}.$$

Due to uniqueness of the local two-sided unit and Proposition 3 we have $E_A = E_{A^i} = A^{t-i}$. Evidently, in the considered sequence there exists some minimum integer $t > i$ which defines some minimum integer $\omega = t - i$ such that $A^\omega = E_A = R_A$. The proposition 4 is proven.

The value ω can be called local order of the vector A. Respectively, the vectors A such that $\Delta_A \neq 0$ can be called locally invertible vectors.

Proposition 5. Suppose the vector A is such that $\Delta_A \neq 0$. Then $A \circ L_i \neq A \circ L_j$, if L_i and $L_j \neq L_i$ are arbitrary two different global left-sided units.

Proof. Suppose $A \circ L_i = A \circ L_j$. Then $A \circ (L_i - L_j) = O$. Since $\Delta_A \neq 0$, the vector equation $A \circ X = O$ has a unique solution $X = O$. Therefore, we have $L_i - L_j = O \Rightarrow L_i = L_j$. The obtained contradiction proves the Proposition 5.

Proposition 6. Suppose the vector equation $X \circ A = B$ has solution $X = S$. Then p^4 different values $X_i = S \circ L_i$, where L_i takes on all values from the set of the global left-sided units (10), also represent solutions of the given equation.

Proof. $(S \circ L_i) \circ A = S \circ (L_i \circ A) = S \circ A = B$. The Proposition 6 is proven.

3 The proposed new form of the HDLP

The known form of the HDLP [12] used in the public-key cryptoschemes is defined in the multiplicative group Γ of some FNAA (finite algebra of quaternions) with global two-sided unit, which is defined over the ground field $GF(p)$, as follows. Two 4-dimensional vectors $Q, G \in \Gamma$ having large prime order ω, which satisfy condition $G \circ Q \neq Q \circ G$, are selected. Then two uniformly random integers $t < \omega$ and $x < \omega$ are generated as the private key and the vector Y is computed as the public key:

$$Y = Q^t \circ G^x \circ Q^{\omega - t} = \left(Q^t \circ G \circ Q^{\omega - t}\right)^x . \tag{18}$$

Finding the pair of integers t and x from equation (18) when the values Q, G, and Y are known is called the HDLP. Finding the value x from the known value G^x, contained in the cyclic subgroup generated by the vector G, represents the DLP, however the value G^x is hidden due to automophism map defined as $Y = TG^xT^{-1}$, where the vector $T = Q^t$ is unknown. The public-key cryptoschemes described in [12] perform correctly due to mutual commutativity of the automophism-map operation and exponentiation operation (see formula (18)). In the paper [13] a potential attack that uses the homomorphism of the group Γ in the multiplicative group of the field $GF(p)$ is proposed. To prevent that attack we propose to define the HDLP in the FNAAs containing no global two-sided unit element (i. e., no globally invertible elements), for example in the 6-dimensional FNAA introduced in Section 2.

In the proposed form of the HDLP the following facts are used: **Proposition 7.** Suppose the product of two 6-dimensional vectors A and B is equal to the global left-sided unit L, i. e. $A \circ B = L$. Then for arbitrary natural number i the equality $A^i \circ B^i = L$ holds true.

Proof.
$A^i \circ B^i = A^{i-1} \circ \left(L \circ B^{i-1}\right) = A^{i-1} \circ B^{i-1} = A^{i-2} \circ B^{i-2} = \cdots = A \circ B = L$.

The Proposition 7 is proven.

Proposition 8. Suppose the product of two 6-dimensional vectors A and B is equal to the global left-sided unit L, i. e. $A \circ B = L$. Then

the map defined by the formula $\psi(X) = B \circ X \circ A$, where the vector X takes on all values in the considered FNAA, is a homomorphism.

Proof. Suppose X_1 and X_2 are arbitrary two 6-dimensional vectors. Then we have

$$\psi(X_1 \circ X_2) = B \circ (X_1 \circ X_2) \circ A = (B \circ X_1 \circ A) \circ (B \circ X_2 \circ A) = \psi(X_1) \circ \psi(X_2);$$

$$\psi(X_1 + X_2) = B \circ (X_1 + X_2) \circ A = (B \circ X_1 \circ A) + (B \circ X_2 \circ A) = \psi(X_1) + \psi(X_2).$$

The Proposition 8 is proven.

Proposition 9. The homomorphism map operation $\psi(X) = B \circ X \circ A$, where $A \circ B = L$, and the exponentiation operation X^i are mutually commutative, i. e., the equality $B \circ X^i \circ A = (B \circ X \circ A)^i$ holds true.

Proof. Due to Proposition 8 we have $\psi(X^i) = (\psi(X))^i$, i. e., $B \circ X^i \circ A = (B \circ X \circ A)^i$. The Proposition 9 is proven.

We propose to use as post-quantum cryptographic primitive the HDLP defined with the following formula for computing the public key:

$$Y = B^t \circ N^x \circ A^t = \left(B^t \circ N \circ A^t\right)^x, \tag{19}$$

where the vector N is such that $\Delta_N \neq 0$, besides the local order of the vector N contains a prime divisor having sufficiently large size.

4 Candidates for post-quantum public-key cryptoschemes

4.1 The public key-agreement scheme

The common parameters of the formula (19) defining computation of the public key Y after selection of the private key (t, x) are generated as follows:

1. Generate a 256-bit prime number $p = 2q + 1$, where q is prime, for example,

$p = 61307880204127427930866981627885239778341924428642533948984609893264740644403;$
$q = 30653940102063713965433490813942619889170962214321266974492304946632370322201.$

2. Generate a random locally invertible vector N having order $\omega = q$.

3. Generate a random locally invertible vector A having order $\omega = q$, which satisfies the condition $A \circ N \neq N \circ A$.

4. Select a random left-sided unit L.

5. Compute the vector B as solution of the vector equation $A \circ X = L$ with the unknown value X.

The first and second users generate their private keys (t_1, x_1) and (t_2, x_2) correspondingly. Then, using the formula (19) they compute their public keys Y_1 and Y_2. The public key-agreement scheme is described as follows:

1. The users exchange their public keys via a public channel.

2. The first user computes the 6-dimensional vector $Z_1 = B^{t_1} \circ Y_2^{x_1} \circ A^{t_1}$.

3. The second user computes the 6-dimensional vector $Z_2 = B^{t_2} \circ Y_1^{x_2} \circ A^{t_2}$.

Correctness proof of the protocol consists in proving that the both users compute the same value Z:

$$Z_1 = B^{t_1} \circ \left(B^{t_2} \circ N^{x_2} \circ A^{t_2}\right)^{x_1} \circ A^{t_1} = B^{t_1+t_2} \circ N^{x_2 x_1} \circ A^{t_2+t_1};$$
$$Z_2 = B^{t_2} \circ \left(B^{t_1} \circ N^{x_1} \circ A^{t_1}\right)^{x_2} \circ A^{t_2} = B^{t_2+t_1} \circ N^{x_1 x_2} \circ A^{t_1+t_2}.$$

Thus, after performing computations in frame of the described cryptoscheme the users share the same secrete value $Z_1 = Z_2$.

4.2 Zero-knowledge protocol

Zero-knowledge protocol is used for authentication of the remote users. It is a public-key method by which one user, owner of the public key, (called the prover) can prove to another one (called the verifier) that he knows the private key connected with his public key, without conveying any information apart from the fact that he knows the private

key. Security of the zero-knowledge protocols is based on the computational difficulty of finding the private key, when the public key is known. Therefore, the post-quantum protocols of such type should be based on the computational problems that are intractable for quantum computers.

To implement a post-quantum zero-knowledge protocol one can use the HDLP described in Section 4 and the method [18] for transforming a public key-agreement scheme into respective zero-knowledge protocol. The idea of such transformation is based on i) ability of two users to compute a common secrete value after exchange of their public keys and ii) applying the single-use public key of the verifier. The proposed protocol is described as follows. The prover is the owner of public key Y computed in correspondence with the formula (19), i. e., he knows the private key (t, x) connected with the vector Y, and the protocol includes the following two steps:

1. The verifier generates a pair of uniformly random natural numbers (g, k) (his single-use private key) and computes his single-use public key D and, using the public key of the prover Y, the value Z:

$$D = B^g \circ N^k \circ A^g; \quad Z = B^g \circ Y^k \circ A^g.$$

Then, using some specified secure hash function F_h, he computes the hash value $h = F_h(Z)$ and sends the values D and h to the prover.

2. The prover computes the value Z' (that is the single-use shared key connected with the public keys Y and D):

$$Z' = B^t \circ D^x \circ A^t.$$

Then he computes the hash value $h' = F_h(Z')$. If $h' = h$, the prover sends the value Z' to the verifier. Otherwise the prover responses "The request is incorrect". The verifier compares the values Z' and Z. If $Z' = Z$, he concludes the prover is genuine. Evidently, the response of the prover contains no information, with exception of the fact that the prover knows the private key (t, x) connected with the public key Y. To ensure that the verifier knows the value of the response Z', Bob checks whether the equality $h' = h$ holds true.

5 Conclusion

The introduced 6-dimensional FNAA does not contain the global two-sided unit, but includes p^4 different global left-sided units. Relatively the lasts one can define the operation of the homomorphism map, which is mutually commutative with the exponentiation operation. The mentioned commutativity and the local invertibility of the elements of the considered FNAA have been used to introduce a new form of the HDLP. The introduced form of the HDLP have been applied to develop the public key-agreement scheme and the zero-knowledge protocol that are proposed as candidates for the post-quantum public-key cryptoschemes. Estimation of their security to different possible quantum attacks appears to be an individual research task. One can suppose that the main approach in frame of such research is connected with the development of the potential quantum attacks on the proposed cryptoschemes which are connected with finding methods for reducing the used HDLP to the DLP in the extension field $GF(p^s)$ with the value $s = 1, 2, \ldots, 6$, like in the case of the HDLP in the finite algebra of quaternions [14].

References

[1] S. Y. Chiou, "Novel Digital Signature Schemes based on Factoring and Discrete Logarithms," *International Journal of Security and its Applications*, vol. 10, no. 3, pp. 295–310, 2016.

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on quantum computer," *SIAM Journal of Computing*, vol. 26, pp. 1484–1509, 1997.

[3] S. Y. Yan, *Quantum Attacks on Public-Key Cryptosystems*, Springer, 2014, 207 p.

[4] *Proceedings of the 7th International Workshop on Post-Quantum Cryptography, PQCrypto 2016, Fukuoka, Japan, February 24-26,*

2016 (Lecture Notes in Computer Science, vol. 9606), 2016, 270 p.

[5] *Post-Quantum Cryptography. 9th International Conference, PQCrypto 2018 Proceedings, Fort Lauderdale, FL, USA, April 9-11, 2018* (Lecture Notes in Computer Science, vol. 10786), 2018.

[6] I. Anshel, M. Anshel, and D. Goldfeld, "An Algebraic Method for Public Key Cryptography," *Mathematical Research Letters*, vol. 6, pp. 287–291, 1999.

[7] E. Lee and J. H. Park, "Cryptanalysis of the Public Key Encryption Based on Braid Groups," in *Advances in Cryptology – EUROCRYPT 2003* (Lecture Notes in Computer Science, vol. 2656), 2003, pp. 477–489.

[8] G. K. Verma, "Probable Security Proof of a Blind Signature Scheme over Braid Groups," *International Journal of Network Security*, vol. 12, no. 2, pp. 118–120, 2011.

[9] P. Hiranvanichakorn, "Provably Authenticated Group Key Agreement based on Braid Groups: The Dynamic Case," *International Journal of Network Security*, vol. 19, no. 4, pp. 517–527, 2017.

[10] A. Myasnikov, V. Shpilrain, and A. A Ushakov, "Practical Attack on a Braid Group Based Cryptographic Protocol," in *Advances in Cryptology ? CRYPTO'05* (Lecture Notes in Computer Science, vol. 3621), 2005, pp. 86–96.

[11] I. Kapovich, A. Myasnikov, P. Schupp, and V. Shpilrain, "Average-case complexity for the word and membership problems in group theory," *Advances in Mathematics*, vol. 190, pp. 343–359, 2005.

[12] D. N. Moldovyan, "Non-Commutative Finite Groups as Primitive of Public-Key Cryptoschemes," *Quasigroups and Related Systems*, vol. 18, no. 2, pp. 165–176, 2010.

[13] D. N. Moldovyan and N. A. Moldovyan, "Cryptoschemes over hidden conjugacy search problem and attacks using homomorphisms," *Quasigroups Related Systems*, vol. 18, no. 2, pp. 177–186, 2010.

[14] A. S. Kuzmin, V. T. Markov, A. A. Mikhalev, A. V. Mikhalev, and A. A. Nechaev, "Cryptographic Algorithms on Groups and Algebras," *Journal of Mathematical Sciences*, vol. 223, no. 5, pp. 629–641, 2017.

[15] N. A. Moldovyan, "Unified Method for Defining Finite Associative Algebras of Arbitrary Even Dimensions," *Quasigroups Related Systems*, vol. 26, no. 2, pp. 263–270, 2018.

[16] A. A. Moldovyan and N. A. Moldovyan, "Post-quantum signature algorithms based on the hidden discrete logarithm problem," *Computer Science Journal of Moldova*, vol. 26, no. 3(78), pp. 301–313, 2018.

[17] A. A. Moldovyan, "General Method for Defining Finite Noncommutative Associative Algebras of Dimension $m > 1$," *Buletinul Academiei de Stiinte a Republicii Moldova. Matematica*, no. 2(87), pp. 95–100, 2018.

[18] N. A. Moldovyan, A. A. Moldovyan, and A. N. Berezin, "On Using Mersenne Primes in Designing Cryptoschemes," *International Journal of Network Security*, vol. 18, no. 2, pp. 369–373, 2016.

Received September 29, 2018

D.N. Moldovyan,

St. Petersburg Institute for Informatics and Automation
of Russian Academy of Sciences,
14 Liniya, 39, St. Petersburg 199178, Russia
E–mail: mdn.spectr@mail.ru; http://www.spiiras.nw.ru

Computer Science Journal of Moldova, vol.**27**, no.1(79), 2019

Predicting the occurrence of strokes using the language R*

Vladimir Popukaylo

Abstract

Probability of stroke is analyzed based on data from Stroke.md system. Selection of features that significantly affect target variable was made. To solve the problem, classification algorithms are used: support vector machines, logistic regression, decision trees, random forest and others; comparison of the resulting mathematical models; preprocessing and building models produced in R-language.

Keywords: mathematical modeling, stroke prediction, data analysis, Stroke.md

MSC 2010: 68R10, 68Q25, 05C35, 05C05.

1 Introduction

This paper solves the problem of predicting stroke occurrence, based on data obtained from Stroke.md information system [1].

At the time of the research, the database contained information about 338 patients, including 96 patients who underwent routine medical examination and 242 patients who were hospitalized with stroke. These data were received while working on the State Program Risk factors, optimizing healthcare service, sustainable assessment and mathematical modeling of stroke [2].

The database contains information about 85 factors, including: personal data, lifestyle information, history, various types of laboratory tests.

* This work was supported by National Agency of Research and Development (NARD) project Ref. Nr. 17.000418.80.07A

To solve this problem, the R language [3] is used with libraries that implement various stages of data preprocessing, building models of classification and visualization of the results obtained. The main stages are implemented using the caret library, which implements a standard approach to machine learning [4].

2 Research problem statement

The research problem consists in creation of predictive model which will allow predicting with a certain share of confidence the probability of developing a stroke at a patient by results of collecting the anamnesis and carrying out some laboratory researches.

To carry out this study, a programming language R was chosen as a language specially created for conducting scientific researches and possessing a large number of additional libraries that facilitate both the preliminary analysis of data and the construction of predictive models.

When forecasting emergence of a stroke at the person it is necessary to solve a problem of classification which in our case is binary. Predictors in the constructed model should be some factors that are stored in the Stroke.md database and significantly affect the probability of stroke.

The solution of the problem can be divided into several stages:

1. Data conversion. At this stage, it is necessary to single out the target variable, to transform some factors into a form convenient for further automated processing, and also to remove non-variable and weakly filled factors from the dataset.

2. Selection of the signs which significantly influence the probability of developing a stroke.

3. Creation of classification models for unbalanced groups and assessment of their efficiency.

3 Data preprocessing

Data for research represent the database dump storing information on the patients who underwent preventive inspection (96 people); and patients who at the time of the research had a stroke in acute phase, divided into two waves of investigation (137 and 105 people).

Thus, the first step was to select the target variable, for which a vector variable was created that stores the value 1 if the patient had a history of any type of stroke before, and also if he had a stroke at the time of the study. At the same time, it turned out that 4 patients who had undergone a preventive examination did not have earlier information whether there had been a stroke, both ischemic and hemorrhagic. It was decided to reject these lines as not-informative. Also there were rejected not-informative columns, such as patients counting number, number of medical card and name of the table from which information was obtained. When analyzing the resulting table, it turned out that there are columns which are filled for less, than 30%. These are:

- Beta-lipoprotein (125 not available definition, hereinafter referred to as NAs).

- Thrombin clotting time (230 NAs).

- Years that a person has not smoked (contains 278 NAs), NAs are replaced by zeros.

- Group of obstetrical background (min. 125 NAs), NAs are replaced by zeros. This is due to the fact that this factor is not filled especially for men.

Thus, two more factors were excluded from consideration.

The following step became reduction of types for factors of initial data set: they were divided into quantitative and qualitative. R is a language with dynamic type-checking, but it does not always correctly recognize the type of a variable, therefore this procedure must be carried out manually. In addition, for qualitative variables, in the case of gradation of factor describing an unknown parameter value, it was replaced by NA. After carrying out these transformations, the parameters were rechecked for missing values and the columns filled for less than 30% were rejected, such as:

- Unruptured aneurysm.
- Metabolic syndrome.
- Valvulopathies.
- Acute myocardial infarction.
- Peripheral vascular disease in family history.

When constructing mathematical models that are based on various forms of data randomization, one has to deal with the problem of missing values. There are three main approaches to its solution:

1. Delete all incomplete lines. In our case, using this approach, we have to reject 244 lines, which is more than half of the available data and it does not seem to be rational.

2. Fill in the missing values using sample statistics of the corresponding variable, assuming that there is no interconnection between the variables. As we cannot suppose that the considered data is distributed according to the normal law, it is preferable to use the median for replacing the missing observations. Such a replacement ignores the specific characteristics of each patient, but it may not be a bad first approximation.

3. To fill the passed values taking into account linear correlation or one of proximity measures. As an example of such approach, the method of the closest neighbors, or intermediate models of regression and classification can be accepted.

The last step in data preprocessing was the removal of columns that store information about previously suffered strokes and near zero variance columns (Ruptured aneurysm, drug related).

After the conversion was completed, a table was obtained containing 334 rows and 71 columns, 30 of which are quantitative and 41 are qualitative.

Thus, an array of initial data was obtained, suitable for further processing by machine learning methods.

4 Feature selection

For selection of features that significantly affect the probability of stroke developing, classical statistical tests were carried out. Thus, for qualita-

tive data, Fisher's two-sided exact test was used as the most powerful of the existing methods, and the Wilcoxon-Mann-Whitney test was used to analyze quantitative data, since the hypothesis about the normal distribution of factors in dataset was not tested. To solve the problem of multiple comparisons, two approaches were used: control of group error probability of the first kind (Bonferroni correction) and control of the average proportion of false deviations (Benjamin-Hochberg method).

This approach allowed us to identify the following factors as significantly influencing the likelihood of stroke (using the Bonferroni amendment):

Qualitative factors are:

- Fasting ($p < 0.0000001$).
- Tooth extraction ($p < 0{,}0000001$).
- Hypertension ($p = 0.00002$).
- Rhythm ($p = 0{,}00034$).
- Marital status ($p = 0.00054$).
- Paradontosis ($p = 0.00383$).
- Occupation ($p = 0.01205$).
- Systemic diseases ($p = 0{,}01237$).
- Atrial fibrillation ($p = 0.01698$).
- Physical activity ($p = 0.02190$).

Quantitative factors are:

- Age ($p < 0.0000001$).
- Pregnancies ($p < 0.0000001$).
- Fibrinogen ($p < 0.0000001$).
- Systolic Blood Pressure ($p = 0.000002$).
- Glucose ($p = 0.00282$).
- Spontaneous abortions ($p = 0.01904$).
- Diastolic Blood Pressure ($p = 0.02101$).
- Leukocytes WBC ($p = 0.03710$).

Usage of less conservative amendment of Benjamin-Hochberg allowed identifying the following parameters in addition: Births; Activated partial thromboplastin time APTT; Sleep Disorders; Address; Sex; Old myocardioal infarction; Migraine.

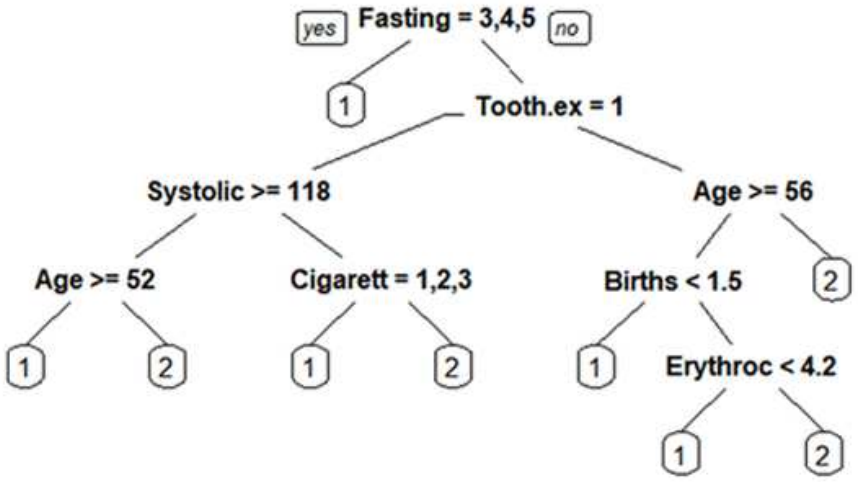

Figure 1. Decision Tree

An alternative method of searching for significant factors is to build a decision tree. So, using the 10-fold cq-cross validation, the following most significant parameters were selected: Fasting, Age, Tooth extraction, Systolic blood pressure, Fibrinogen, Births, Hemoglobin, Activated partial thromboplastin time APTT, Pregnancies, Sex, Dental prosthesis, Creatinine, Paradontosis, Ocupation, Height, Cigarette smoking, Hypertension, Antihypertensive, Medical abortions, Hematocrit HCT, Erythrocytes RBC, ASAT, Diastolic blood pressure, Marital status, Leukocytes WBC, Cholesterol.

Figure 1 shows the image of one of the constructed trees.

The considered algorithms of searching the factors which are significantly influencing a target variable allow identifying a number of the factors requiring closer attention at prevention of strokes.

It should be noted that the list of factors selected by each of the methods is somewhat different from that generally accepted in the lit-

erature [5] on this topic.

This could depend on the characteristics of the data collected. However, our task was to develop a model based on the available information, so it was decided to use the above columns as predictors.

5 Creation of predictive models

For training the models and selecting their hyperparameters, the caret library was used, which provides universal interface for training models of various types.

At the stage of selecting the structure of the model, a 10-fold repeated cross validation was carried out. Different types of models have been tested, which are: logistic regression, decision trees, support vector machine and various boosting algorithms. Figure 2 shows that algorithms based on the construction of model ensembles show more reliable results, both in accuracy prediction and in agreement between classes.

The best results were obtained using such models as: Adaboost.M1; Random Forest; DeepBoost; Extreme Gradient Boosting on Trees; Boosted Generalized Linear Model; Extreme Gradient Boosting on Linear models.

Table 1 presents the numerical values for the Accuracy parameter, obtained by cross-validation for some of the algorithms.

Most of the known boosting models based on the base models of various types show statistically indistinguishable results with a median accuracy 0.8261. At the same time confidential intervals show insignificant advantage of such algorithms as Adaboost for decisions trees models and Extreme Gradient Boosting for linear models.

Comparison of coherence between classes on Kappa Kohen's criterion also confirms previous findings. At the same time, the average consistency of classes for the specified algorithms varies at a level of 0.5, which indicates satisfactory agreement.

It is known that adaptive boosting models can screen out insignificant parameters, however, in the case of supersaturated tables, the algorithms may not give the best indicators.

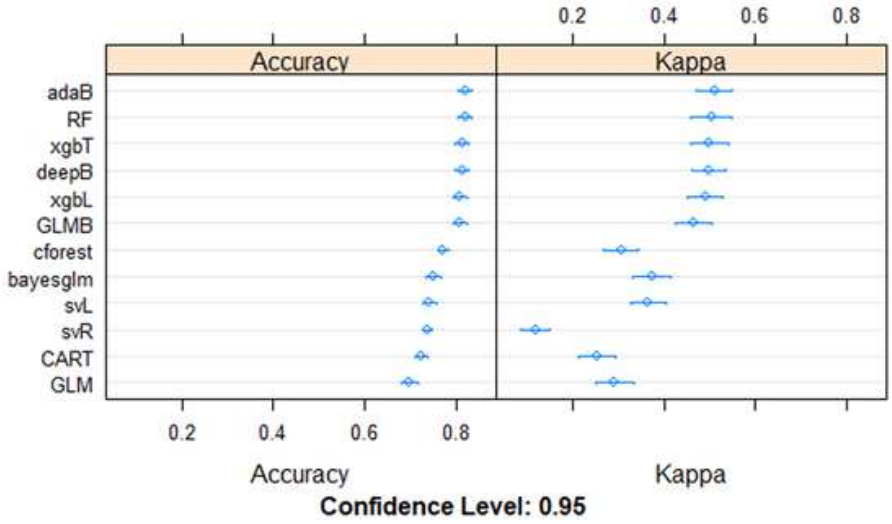

Figure 2. Accuracy and Kappa for models with all predictors

Table 1. Accuracy of models with all predictors

	$Min.$	$1st.Qu.$	$Median$	$Mean$	$3rd.Qu.$	$Max.$
$adaB$	0.6522	0.7826	0.8130	0.8193	0.8696	1.0000
$cforest$	0.5833	0.7391	0.7826	0.7719	0.8261	0.9130
$GLMB$	0.6667	0.7500	0.7917	0.8086	0.8696	0.9583
$BGLMB$	0.4800	0.6957	0.7500	0.7513	0.8261	0.9167
$deepB$	0.6522	0.7500	0.8261	0.8130	0.8696	0.9583
$CART$	0.5417	0.6667	0.7391	0.7243	0.7826	0.8750
RF	0.6087	0.7500	0.8261	0.8191	0.8750	1.0000
GLM	0.4783	0.6522	0.7083	0.6978	0.7500	0.8800
$svmL$	0.4167	0.6957	0.7391	0.7414	0.7917	0.9167
$svmR$	0.6087	0.7083	0.7391	0.7360	0.7826	0.8333
$xgbT$	0.6522	0.7500	0.8261	0.8132	0.8696	0.9583
$xgbL$	0.6250	0.7826	0.8261	0.8088	0.8696	0.9565

In this connection, it was decided to train the models with the most important selected parameters. The best results were obtained when selecting significant features for statistical tests with a significance level of p <0.05 and a Benjamin-Hochberg correction based on a false discovery rate.

The Accuracy values received on cross-validation for the algorithms which improved the indicators are presented in Table 2.

Table 2. Accuracy of models with 25 predictors

	$Min.$	$1st.Qu.$	$Median$	$Mean$	$3rd.Qu.$	$Max.$
$adaB$	0.6522	0.7826	0.8333	0.8292	0.8750	1.0000
$cforest$	0.6250	0.7391	0.7826	0.7863	0.8474	0.9130
$GLMB$	0.6522	0.7826	0.8261	0.8223	0.8696	0.9565
$BGLMB$	0.6087	0.7500	0.8130	0.8078	0.8696	1.0000
$deepB$	0.6522	0.7826	0.8333	0.8275	0.8750	0.9583
$CART$	0.5417	0.6957	0.7500	0.7483	0.7917	0.9200
RF	0.6957	0.7917	0.8261	0.8291	0.8696	0.9583
GLM	0.5833	0.7473	0.7917	0.7972	0.8696	0.9583
$svmL$	0.6250	0.7391	0.7917	0.7911	0.8333	0.9565
$svmR$	0.6522	0.7826	0.8261	0.8203	0.8696	1.0000
$xgbT$	0.6667	0.7917	0.8696	0.8521	0.9130	1.0000
$xgbL$	0.6522	0.7917	0.8514	0.8427	0.9130	1.0000

The confidence intervals for the Accuracy and Kappa parameters are presented in Figure 3.

Thus, it can be seen on Figure 3 that the best results at the stage of cross-qualification were shown by the Extreme Gradient Boosting on Tree algorithm (xgbT). This algorithm showed the best tune with hyper parameters:

- eta = 0.4;
- nrounds = 150;
- max depth = 3;
- colsample by tree = 0.6.

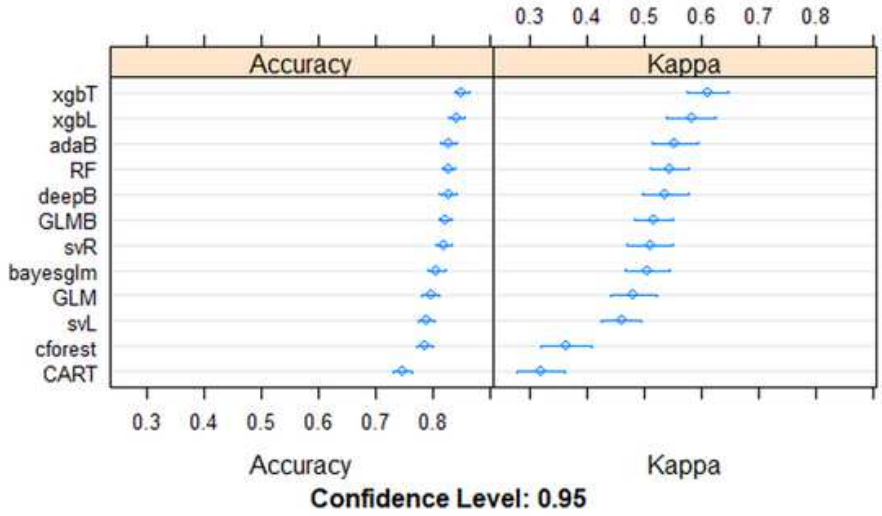

Figure 3. Accuracy and Kappa for models with 25 predictors

Among non-boosting models, the best results were obtained using support vector machine with radial kernels.

At the next stage, the accuracy of the selected algorithms for the holdout dataset (99 items) was evaluated. The results obtained for the models that showed the best results are presented in Table 3.

6 Conclusions

During the conducted research it was succeeded to improve the results received earlier [5]: to construct the predictive models capable with an approximate accuracy at the level of 80% to classify patients who had stroke; parameters which can influence the occurrence of a stroke were selected. The best quality in forecasting was reached by means of the algorithm of extreme gradient boosting on trees of decisions. Median accuracy for this algorithm at a stage of cross-validation equals 86.96%,

Table 3. Accuracy of models for the holdout dataset

Algorithm	*Accuracy*
Extreme Gradient Boosting on Decision Trees	0.77778
Random Forest	0.79798
Boosted Generalized Linear Model	0.78788
Extreme Gradient Boosting on Linear Model	0.78788
Support Vector Machine with Radial Kernel	0.79798
Support Vector Machine with Linear Kernel	0.75757
Generalized Linear Model	0.74747
Bayes Generalized Linear Model	0.77778
*AdaBoost.M*1	0.74747
DeepBoost	0.72727

on the holdout dataset accuracy equals 77.78%. Among the models which are not based on randomization, the best results were shown by support vector machine with radial kernel. Median accuracy of such model at a cross-validation stage – 82.61%, on the holdout dataset – 79.8%.

The results can be used in the software systems development targeted at preventing strokes among the population.

References

[1] E. Zamsa, "Medical software user interfaces, stroke MD application design," in *2015 E-Health and Bioengineering Conference – EHB*, 2015, pp. 1–4.

[2] S. Groppa, N. Ciobanu, D. Efremova,"Stroke risk factors in the population of Republic of Moldova," *Journal of the Neurological Sciences*, vol. 381, p. 411, Oct., 2017.

[3] M. Kuhn, "Building predictive models in R using the caret package," *Journal of statistical software*, vol. 28, no. 5, pp. 1–26, Nov., 2008.

[4] RC Team. *R language definition* (2019) [Online]. Available: ftp://155.232.191.133/cran/doc/manuals/r-devel/R-lang.pdf

[5] S. Cojocaru *et al.*, "Analysis and preparation of data from Stroke.md database when creating a stroke prediction model," in *Proc. MFOI-2018*, 2018, pp. 51–66.

Vladimir Popukaylo Received March 18, 2019

Vladimir Popukaylo
Vladimir Andrunakievich Institute of Mathematics and Computer Science
T.G. Shevchenko Transnistrian State University
Moldova, Tiraspol, str. Vosstania 2a, cab. 311B
Phone: +37377844001
E–mail: vsp.science@gmail.com

Computer Science Journal of Moldova, vol.**27**, no.1(79), 2019

The impact of parameter optimization of ensemble learning on defect prediction

Muhammed Maruf Öztürk

Abstract

Machine learning algorithms have configurable parameters which are generally used with default settings by practitioners. Making modifications on the parameters of machine learning algorithm is called hyperparameter optimization (HO) performed to find out the most suitable parameter setting in classification experiments. Such studies propose either using default classification model or optimal parameter configuration. This work investigates the effects of applying HO on ensemble learning algorithms in terms of defect prediction performance. Further, this paper presents a new ensemble learning algorithm called novelEnsemble for defect prediction data sets. The method has been tested on 27 data sets. Proposed method is then compared with three alternatives. Welch's Heteroscedastic F Test is used to examine the difference between performance parameters. To control the magnitude of the difference, Cliff's Delta is applied on the results of comparison algorithms. According to the results of the experiment: 1) Ensemble methods featuring HO performs better than a single predictor; 2) Despite the error of triTraining decreases linearly, it produces errors at an unacceptable level; 3) novelEnsemble yields promising results especially in terms of area under the curve (AUC) and Matthews Correlation Coefficient (MCC); 4) HO is not stagnant depending on the scale of the data set; 5) Each ensemble learning approach may not create a favorable effect on HO. To demonstrate the prominence of hyperparameter selection process, the experiment is validated with suitable statistical analyzes. The study revealed that the success of HO which is, contrary to expectations, not depended on

the type of the classifiers but rather on the design of ensemble learners.

Keywords: Defect prediction, parameter optimization, ensemble learning.

1 Introduction

An intensive effort is devoted to defect prediction which helps practitioners to plan the budget of a software engineering project [1]. Due to the advance in software systems, including limited budgets, much works are required in this domain. Defect prediction works are generally focused on software metrics [2], predictors [3], pre-processing methods [4], and prediction models [5]. However, the works that explore which type of classifiers should be used in data sets having a broad scale of range are intriguing for researchers [6]. In addition to this, in recent years, predictor configuration studies, which aim to find appropriate settings of the predictors, have emerged [7]–[10]. They open new horizons for the development of enhanced prediction models.

Defect prediction is a process which aims to predict future defects of a software project by using historical metrics of related data sets along with defectiveness information. It is generally considered as a binary classification problem. In defect prediction, high error rates are decreased by combining more than one classifier. Predictors could make different decisions about the label of an instance depending on its structural properties. Thus, each predictor yields the best results in different data sets. Regarding this case, various ensemble learning approaches have been developed. Combining more than one classifier to solve a computational problem is generally named ensemble learning [11]. It is employed for improving the performance of a model such as clustering, classification, and approximation. Ensemble learning methods generally adopt three techniques. These are stacking [11], bagging [12], and boosting [13]. Figure 1 demonstrates how an instance is labeled through ensemble learning techniques.

Generally, an ensemble learning method first selects the classifiers to be employed in the method. To this end, the number of the clas-

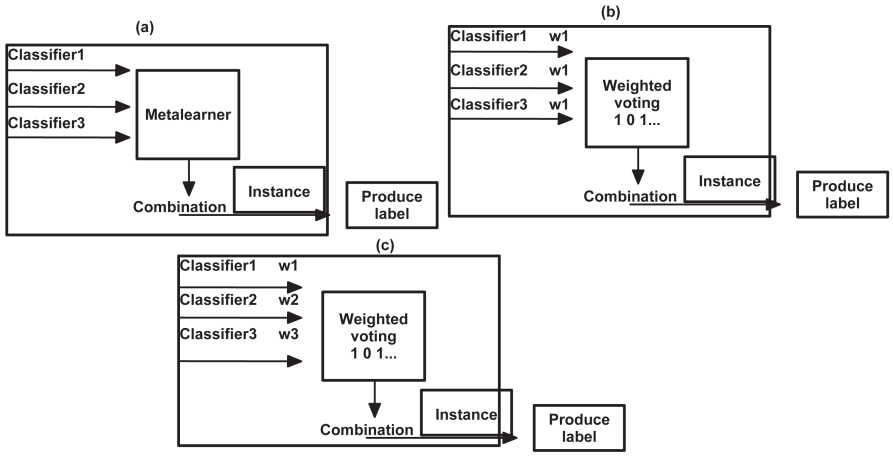

Figure 1. Three different approaches for ensemble learning. (a) Stacking: Uses a meta-learner to decide the label of instance, (b) Bagging: Gives equal weights to the classifiers, (c) Boosting: Utilizes the same mechanism with Bagging but it gives different weights to the classifiers.

sifiers should be determined in the beginning. The performance of a classifier strongly depends on the type of the data sets. Enormous software systems give large-scale defect prediction data sets to practitioners. Therefore, practitioners should work with the classifiers that are suitable for large-scale data sets. Thereafter, a design is established depending on the error rates and performance parameters. In this design, the number of the classifiers and the settings of hyperparameters are determined in accordance with experimental knowledge.

One of the important points that have long intrigued experts in ensemble learning is how to select hyperparameter settings [14]. Default settings of hyperparameters may not produce consistent results depending on the used data sets. Rather, employing ensemble models generated with various hyperparameters which are determined according to the classifier and the data sets may be a better choice. In the meantime, tuning hyperparameters is relatively time-consuming and effort-intensive process comparing to the traditional defect prediction [7].

With respect to the ensemble learning approaches, the number of classifiers is not entirely standard in the existing methods [15], [16]. Further, it is also ambiguous that which types are to be selected from a great number of predictors. To make a good selection, each predictor should be evaluated for a general hyperparameter configuration.

Various challenges are encountered while using an ensemble learning method. The most notable of them is whether default settings of the classifiers are adopted or not [17]–[19]. Generally, default configuration of the classifiers is used in an ensemble learning method. Preferring such a way degrades the reliability of an experiment. Conversely, exploring hyperparameter settings by testing previous assumptions may yield better results with regard to the proposed method.

HO was examined in terms of defect prediction in a few works [7], [8], [10]. However, the effects of HO on ensemble methods have not been investigated yet. This case can be considered as an obstacle that should be removed to fully comprehend HO. Moreover, whether HO improves overall performance of advanced classifiers is an open issue.

Cross-project defect prediction (CPDP) is one of the main long-standing problems of defect prediction [20], [21]. In this type, training and testing data are taken from different software projects. On the other hand, within-project defect prediction (WPDP) aims to perform prediction by taking different data sets of the same project so that there is not any metric contradiction problem thanks to the use of same metric sets.

In our preceding work, the effects of HO were investigated in CPDP and WPDP [22]. It was intended to provide valuable findings to find out whether HO increases the success of the prediction when heterogeneous metrics are employed. The method was tested on 20 data sets with heterogeneous experimental setting. As a result, it was detected that performance values of WPDP tend to be in a wide range. With respect to AUC and F-measure parameters, if the classifiers working with tree-structured data are employed, CPDP is capable of yielding feasible performance values in which the depth of trees is high.

Main objectives of the paper are as follows: 1) Investigate the effects of applying HO on three ensemble learning methods; 2) Develop a

robust ensemble learning method that results in performance increase when HO is utilized; 3) Provide a new insight into defect prediction in terms of HO. In addition, this paper aims at developing a novel ensemble learning method called novelEnsemble which is expected to produce promising results when HO is employed.

To achieve those goals, open-source data sets are collected in the beginning of the experiment. In the second step, some ensemble learning and HO methods are selected. Third step encompasses developing novelEnsemble in accordance with constructive methods. In the last step, performance results obtained from both applying HO and without HO on ensemble learning methods are recorded. Meanwhile, the most known ensemble learning methods are tried to be evaluated with general hyperparameters. Main steps of the experiment are demonstrated in Figure 2. As known from the figure, data sets are initially divided into two parts. Subsequently, data sets are exposed to normalization and exploited by the algorithms in which default and optimized parameters are used. Last, some performance parameters are yielded and evaluated in the test process.

The contributions of the paper can be summarized as follows:
1) A novel ensemble learning method namely novelEnsemble, which uses mean classification error and total number of defects for instance labeling, is proposed.
(2) This paper reveals which properties should an ensemble learning method have when HO is applied
(3) Obtained findings of the experiment shows that the type of the parameter search is negligible in HO performance.

The remainder of the paper is organized as follows: In Section 2, the motivation and the shortcomings of previous works are elaborated. Section 3 refers to ensemble learning and HO along with formal definitions. Related works and ensemble learning methods associated with the paper are also in Section 3. Section 4 presents the proposed method. Section 5 describes the data sets and experimental environment. Experimental results are given in Section 6. Threats for the validity are given in Section 7. Last, the findings and the outcomes of the paper are presented in Section 8.

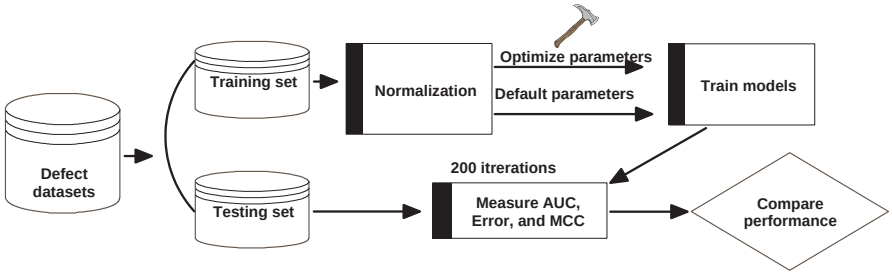

Figure 2. Main steps of the study.

2 Motivation

This section provides four motivations for the experiment. Motivations of the paper are elaborated by giving some references to draw a clear outline.

HO studies in machine learning algorithms are not new in defect prediction [23]–[25]. Their main objective is to find optimal parameters of learning algorithms rather than using default configuration. They were able to produce promising results in terms of HO. However, due to a great number of hyperparameters and classifiers, practitioners who perform hyperparameter selection feel confused [10]. Therefore, bringing a new perspective to HO in terms of classifiers constructing ensemble learner is the first motivation.

Defect prediction is generally a binary classification problem. Therefore, it has configurable settings. HO can be considered as a new research field in terms of defect prediction. Preceding works have investigated hyperparameter search method [10], automated hyperparameter optimization [8], and the effects of HO on defect prediction [7]. However, with respect to the defectiveness indicators of defect prediction data sets, the effects of HO in the classification comprising more than one classifier have not been fully investigated up to this study. Second motivation is to find optimal design of an ensemble learner to yield high classification performance.

On the other hand, ensemble learning methods are preferable to

increase the success of defect prediction [26], [27]. Note that an ensemble learner cannot be devised arbitrarily. The number of classifiers and used approach including bagging, boosting, and stacking do affect the results of ensemble learning method. In addition, configuration of hyperparameters plays an important role in the performance of the prediction. Previous works have pointed out that each classifier may not be compatible with ensemble learning. Further, some classifiers such as C5.0 and neural network tend to produce high performance [8]. Comparing classifiers in contributing the performance of an ensemble learner is the third motivation.

Ensemble learning methods were developed on the basis of pioneer methods such as triTraining [28]. However, in such methods, instance labeling is focused on classification error and the number of classifiers is constant. Instead, practitioners need a novel ensemble learning method which does not depend on the number of classifiers and can be devised by considering the properties of defect prediction data sets. Constructing an ensemble learner by regarding common properties of defect prediction data sets is the fourth motivation.

3 Background and related works

3.1 Preliminaries

If a software is represented with various modules $s_1, s_2, s_3, ..., s_n$, a set of modules are labeled with $l_1, l_2, l_3, ..., l_n$. These labels are either "defective" or "not-defective". A classifier is trained on some parts of labeled instances to predict unlabeled testing instances. This operation is done with a classifier c. In some cases, the number of classifiers may be higher than one $c_1, c_2, c_3, ..., c_m$. In such experiments, classifiers are unified by ensemble techniques for deciding the labels of testing instances.

To perform ensemble learning, a classifier c is weighted according to the type of learning. One could assume that $w_1, w_2, w_3, ..., w_m$ are the weights of $c_1, c_2, c_3, ..., c_m$. To decide a label l of s, classifiers make a bias set such as $1, 0, 1, 1, 1, ...$ That set is combined with the weights

$w_1, w_2, w_3, ..., w_m$. For a binary classification problem, an ensemble learner produces an output y as 1 or 0 in compliance with the type of learners. In this process, the classifiers used in constructing the ensemble learner and the type or the data sets are of great importance. An underlying deciding mechanism of an ensemble learner is given in Figure 3.

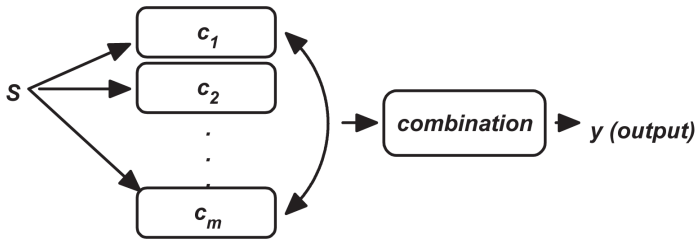

Figure 3. An underlying deciding mechanism of an ensemble learner. The learners are combined by giving a specific set of weights to create a combination for producing an output.

In testing, a set of instances to be predicted can be denoted with $st_1, st_2, st_3, ..., st_p$. p denotes the number of testing instances. However, it is not compulsory to equalize p and n. Instead, p is generally determined as less than n. For this purpose, a specific percentage is used to perform testing. Otherwise, cross-validation methods are applied. 10-fold cross-validation is one of the most popular of them.

3.2 Related Works

Related works can be examined in twofold: these are ensemble learning and HO studies which are closely related to defect prediction. In this respect, existing works have been summarized and the need for performing the experiment has been stressed. Due to the fact that HO is not a new research field for defect prediction, the number of related studies is limited. However, prevalence of HO in search-based software engineering has helped to solve this issue.

The efficacy of ensemble learning approaches depends on the prop-

erties of defect prediction data sets in a specific ratio. In a study [26] where the prediction performance is examined by combining feature selection with ensemble learning, forward selection outperformed other feature selection techniques. It can be considered one of the basic studies because its findings revealed that feature selection is of great importance for ensemble learning. In another study [29], feature selection helped to solve class imbalance problem and increased the success of ensemble learning. Such hybrid models had been tested in some areas except defect [27]. For instance, a hybrid model was tested on banking system and yielded promising results [27].

Wang et. al proposed a new classifier for defect prediction data sets by benefiting the advantages of kernel and ensemble learning approaches [6]. Although the method produced high results in F-measure, employing only industrial data sets without considering open-source ones creates a crucial threat for the validity. Further, a tuning operation was not conducted during the classification experiment.

The success of ensemble learning was also evaluated for multi-class defect prediction. In one of them [30], initially, binary-labeled instances were converted to multi-class form. Thereafter, the classification was completed based on a sophisticated coding. Despite this method coped with imbalanced data sets, hyperparameter settings were not considered while developing ensemble learning method. The interaction between class imbalance and ensemble learning much intrigued researchers. In [31], two boosting methods were discussed on five performance measures. According to the obtained results, AdaBoost.NC showed best performance in general.

Laradji et al. proposed an ensemble learning module [26]. Their experiment using seven classifiers depicted that greedy selection is much preferable for feature selection. Although AUC results are close to 1 in three data sets, the method should be verified with different feature selection and ensemble methods. Likewise, Peng et al. pointed out that AdaBoost provides some advantages along with analytical hierarchy process.

Researchers have long strived to determine how a modification made on a software influences defect ratio. In [32], this impact was investi-

gated in terms of scale and proposed a two-layer ensemble method utilizing both bagging and stacking. The method was evaluated with three alternatives on six data sets. It revealed up to 70% of defect-prone modifications. Related paper stated that improving the parameters of proposed method was planned.

HO is one of the focuses of search-based software engineering [33]–[35]. Parameter tuning has long been a topic of interest in the study of search-based software engineering.

Fu et. al claimed that tuning process of defect prediction requires too much time [10]. In their study, it was detected that HO is feasible and increase the accuracy of the prediction up to 60%, especially in defect data sets. While making tuning on selected parameters of the predictors, the effect of parameter search method is not negligible. Thus, a study [9] asserts that parameter search method strongly depends on the scale of the data sets. Moreover, random search method is much suitable with small-scale data sets.

One of the most comprehensive studies was conducted by Tantithamthavorn et al. [8] who examined almost all parameter search methods exhaustively. They concluded that some classifiers such as C5.0 and neural network are rather compatible for HO. It was also stressed that HO should be tested in terms of ensemble learning method.

Practitioners should employ HO without applying a machine learning model, asserted in a recent study [7]. K-nearest neighbor outperformed competing methods with the maximum enhancement in the accuracy of the prediction.

3.3 Ensemble learning algorithms

Ensemble learning approach is an interesting topic that emerged at the end of 90's [36]–[38],[38]. As shortly stated in Section I, three different approaches are adopted in ensemble learning: boosting, bagging, and stacking. Therefore, related experiments are devised to encompass at least one of them. Uncommonly, a study could benefit more than one approach.

In the beginning, software defect prediction had relied on tests performed with one classifier [39], [40]. Ensemble learning studies were revealed after discussing the findings of preceding works [41]. Due to ensemble learning methods date back to the mid of the 1990s, basic methods were applied and their improved versions were produced. In this section, primary ensemble learning methods are elaborated.

Pioneer works preferred to use neural network in constructing ensemble learning methods [42], [43]. Experimental data sets are mainly associated with image processing.

Learn++ is an ensemble learning method which uses lazy classifiers and is fast to neural networks [44]. In Learn++, last decision mechanism is weighted voting. The method converges well as it does not need prior training data instances. Ensemble learning was also utilized in training process. In [45], a learning machine namely EN-ELM was developed and tested on some images. This study depicts the prominence of the selection of parameters for alleviating the computational cost of classification.

Consistent methods can be developed by combining various ensemble learning strategies. WAG is one of them [46] and it was constructed by unifying wagging and bagging. Unlike bagging, wagging assigns weights to training instances using Poisson distribution. The method was able to reduce test error remarkably.

Ensemble learning was also used in the classification of noisy data sets [47]. Thus, a model having high error tolerance and accuracy can only be obtained by that way.

While developing an ensemble learning method, having high number of main classifiers increases computational cost quadratically. Therefore, classifier count is tried to be optimal magnitude. For instance, triTraining is an ensemble method that consists of three classifiers in which classification errors are compared to decide a label of an instance [28]. Test operation is performed by comparing error ratios of three classifiers to decide instance label. triTraining showing better performance than three competing methods is regarded as a reference model to various ensemble learning methods.

In [48], a fuzzy cluster-based ensemble learning approach namely

IFCESR was proposed. It employs soft clustering techniques to create ensemble clusters. The effectiveness of the method was tested on UCI data sets. According to the obtained results, IFCESR surpassed state of the art alternatives in terms of clustering accuracy.

Customer scoring is an interesting field in which ensemble learning was utilized [27]. The method using hybrid methods simultaneously has better performance results with AdaBoost than other methods. Moreover, PCA is much feasible for feature selection rather than information gain and GA. Fuzzy cognitive map was improved with ensemble approach [49]. In doing so, it was observed that the performance of fuzzy cognitive map decreases remarkably when it is employed with Hebbian learning.

Pratama et al. presented a new ensemble learning method namely pEnsemble [50]. It consists of three components: drift detection, ensemble pruning, online feature selection. The main advantage of pEnsemble is that it features less complexity than its alternatives.

3.4 Limitations of existing approaches

Existing works have investigated various aspects of applying HO on defect prediction data sets so far [7]–[10]. They rely on specific assumptions. First, default configuration of classifiers does not yield high success according to the HO models. Thus, a special configuration is searched for each classifier. Second, it is asserted that parameter search method should be selected in accordance with the experimental data sets. In this respect, parameter search methods are compared in varying type of data sets. The common aspect of preceding works is that they do not include any evaluation of HO in terms of ensemble learning approaches. In particular, preceding works lack investigation of ensemble learning approaches in which HO is applied on constructor classifiers. Rather than establishing such an experimental environment, detail information is presented by extending the evaluation interval of hyperparameters. In this case, the experiment is enriched according to the main objective but the scope of the evaluation is restricted.

Ensemble learning approaches date back to older times then HO [6], [26], [47], [51]. Therefore, with respect to the defect prediction, the

number of ensemble studies is more than those of HO. Ensemble learning methods are generally proposed to address class imbalance and noisy data problems. However, mathematical models related to the defectiveness metrics are rarely employed in the construction of an ensemble learning method.

To alleviate the limitations mentioned above, this paper investigates the effects of HO in ensemble learning approaches. To this end, some promising classifiers such as C5.0 and neural network have been used in constructing experimental ensemble methods. In addition to this, novelEnsemble having no restriction on the number of classifiers is proposed. The method considers the number of defects depending on the number of line of codes while determining an instance label.

Performing the experiment for a data set could fill the gaps as follows: 1) Observing the success change of ensemble learning approaches in case of applying HO; 2) Encouraging the development of ensemble methods that are designed for defect prediction rather than general data mining operations; 3) Determining whether each HO operation has a profound impact on defect prediction; 4) Revealing the compatibility level of ensemble learning approaches on HO in which static code metrics are exploited.

4 Method

In the concept of ensemble learning, the main purpose is to combine various classifiers to increase prediction performance. The reason is that using only one classifier in a prediction experiment may not perform well in every experimental condition.

The method presented in Algorithm 1 is a defect prediction algorithm. It can be consisted of various classifiers depending on the experimental design. In this study, Algorithm 1 comprises three classifiers RandomForest, C5.0, and neural network. The error mentioned in the following paragraph is the percentage of incorrectly classified instances in the prediction. The algorithm utilizes defectiveness rate while comparing the errors of the classifiers. The classifier producing the minimum error in the iteration determines the label of an instance in testing.

Algorithm 1 takes three parameters including O, U, and C. This method includes an error additive model. First loop of the algorithm encompasses $S_i = BootStrap(O)$ and $E_i = 0.5$ which is an arbitrary error value that is used if related classifier does not change in terms of error rate during the learning.

In the second loop, N denotes the number of the instances to be exposed to ensemble learning. $E_1 = C_1(U_i), E_2 = C_2(U_i), E_3 = C_3(U_i)$ calculates the error rates of three classifiers. $total \leftarrow total + E_j$ calculates the total error up to the reached instances. $total$ changes increasingly until all the instances are exposed to training. In the latter step, mean error is calculated and assigned to av. Mean error is associated with the number of instances and the number of classifiers. i represents the index of an instance. In the third step, the classifier having minimum error is detected with $z \leftarrow min(e_1, e_2, e_3)$. $y \leftarrow \int av + x^2 + c$ is a function that relies on total defect, mean error, and the number line of codes av, x, c. Since a defective region can be detected with at least two curves as in Figure 4, an integral function is utilized in Algorithm 1.

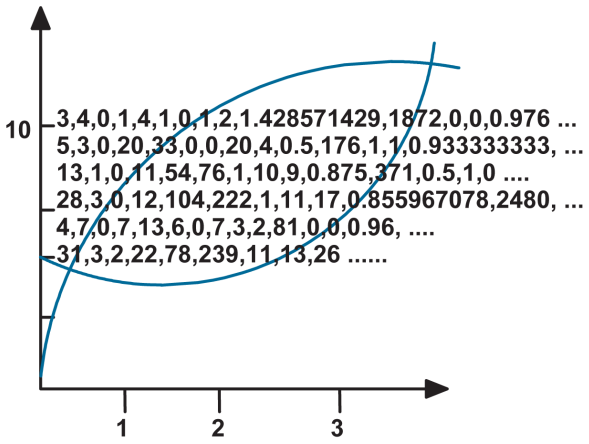

Figure 4. An example illustration of area between curves determining defective region of a defect prediction data sets.

The ambiguousness of the boundary of defective region reveals the need for integral calculation. $(z <= y)$ compares the minimum error obtained in the related instance with the average error of defective region. If the error of the classifier is less than average error, the classifier is used for labeling U_i. Otherwise, C_1, C_2, C_3 calculates the number of labeling biases along with a straight bagging. According to this calculation, testing is completed.

Calculating S_i in Algorithm 1 is similar to that of triTraining. However, the criteria used while deciding the label of an instance are different. For instance, during examining the individual errors of the instances, novelEnsemble is interested in produced errors until mean error and related training instance are in use.

Algorithm 1 novelEnsemble Algorithm

1: **Input** O:Original labeled instance set,
   U:Unlabeled instance set,
   C:RandomForest, C5.0, NeuralNet
2: **for** each item i in C_i **do**
3: $S_i = BootStrap(O)$
4: $C_i = Learn(S_i)$
5: $E_i = 0.7$
6: **end for**
7: **for** each item i in N **do**
8: $E_1 = C_1(U_i), E_2 = C_2(U_i), E_3 = C_3(U_i)$
9: total←total+E_i
10: av←total/i
11: z←min(E_1,E_2,E_3)
12: y←$\int av + (total)^2 + c$ //av=mean error, x=i. total error, c=total defects/line of code
13: if $(z <= y)$
14: $U_i \leftarrow C_z$
15: else $U_i \leftarrow$ compute(C_1, C_2, C_3)
16: **end for**

5 Experiment Design

This section presents experimental data sets and research questions (RQ). Further, experimental settings and performance parameters are elaborated. Last, research questions are discussed.

5.1 Data sets

27 data sets have been used to evaluate novelEnsemble. They include the versions of following projects: ant [52], berek, camel, ckjm [53], e-learning, ivy, jedit, kalkulator, log4j, lucene, nieruchomosci, tomcat, and xalan [54]. These data sets have been obtained from tera-promise repository.

Some properties of experimental data sets are given in Table 1. These projects are open-source and coded with java programming language. An instance denotes either a software module or a class. The number of the instances in the projects is rather different. For instance, while tomcat has 858 instances, berek has 70 instances. Employing various instances in comparison projects alleviates the burden in evaluating the results of the experiments. Further, it ensures a clear bias about novelEnsemble. Table 2 gives used metrics with their definitions and types. There are 24 software metrics. 4 out of 24 metrics are of process metrics.

5.2 Performance Parameters

Performance parameters used in a defect prediction experiment change depending on the type of the problem. For example, if a method is developed for class imbalance, PF, g-mean, and AUC are frequently involved in the evaluation [55], [56]. It is well known that the works including HO have a great number of parameters. However, in recent works, MCC has become popular among researchers handling with defect prediction [57], [58].

In this study, AUC, MCC, and classification error have been used for evaluating novelEnsemble. Due to the main focus of the paper is ensemble learning methods, the scale of the performance parameters

Table 1. Details of the projects used in the experiment.

Project	Version	Number of instances	Number of defects	Defectiveness (%)
ant	1.7	745	338	22
arc	1	234	33	11
berek	1	70	33	43
camel	1.0	339	14	3
camel	1.2	608	522	41
camel	1.4	872	335	16
camel	1.6	965	188	19
ckjm	1.8	10	23	50
e-learning	1	64	9	13
ivy	1.1	111	233	56
ivy	1.4	241	18	6
ivy	2.0	352	56	11
jedit	3.2	272	382	90
jedit	4.0	306	226	24
jedit	4.1	312	217	25
jedit	4.2	367	106	13
jedit	4.3	492	12	2
kalkulator	1	27	7	25
log4j	1.0	135	61	25
log4j	1.1	109	82	33
log4j	1.2	205	498	92
lucene	2.0	195	268	46
lucene	2.2	247	413	57
lucene	2.4	340	630	59
nieruchomosci	1	27	13	37
tomcat	6	858	114	8
xalan	2.4	723	155	15

Table 2. Metrics of experimental data sets.

Name	Description.	Type
wmc	Weighted Methods per Class	Static code
dit	Depth of inheritance	Static code
noc	Number of children	Static code
cbo	Coupling between objects	Static code
rfc	Response for a class	Static code
lcom	Lack of cohesion	Static code
ca	Afferent coupling	Static code
ce	Efferent couplings	Static code
npm	Number of Public Methods	Static code
lcom3	A variant of lcom	Static code
loc	Line of codes	Static code
dam	Data Access Metric	Static code
moa	Measure of. Aggregation	Static code
mfa	Measure of functionality abstraction	Static code
cam	Cohesion Among Methods of class	Static code
ic	Inheritance Coupling	Static code
cbm	Coupling between Methods	Static code
amc	Average Method Complexity	Static code
nr	Number of revisions	Process
ndc	Number of distinct committers	Process
nml	Number of modified lines	Process
ndpv	Number of defects fixed in previous version	Process
max_cc	Maximum Class Coupling	Static code
avg_cc	Average Class Coupling	Static code

is not large. Moreover, to the best of our knowledge, this study is first to discuss ensemble learning in defect prediction along with HO. Therefore, the initiative is made rather than focusing on the detail of the results.

In Table 3, performance parameters used in the experiment are presented. MCC consists of confusion matrix members and AUC is the area under the ROC. In the formula indicating AUC, m are the data points and i refers to the execution number on m data points that denotes true label [59]. On the other hand, j refers to the execution time of n data points. The formula of AUC in each iteration produces *1* if $p_i > p_j$. p denotes the probability value assigned by the classifier to

Table 3. Performance parameters used in the experiment.

name	**formula**
MCC	$\frac{(TP*TN-FP*FN)}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}}$
AUC	$\frac{1}{mn}\sum_{m}^{i=1}\sum_{n}^{j=1} 1_{p_i>p_j}$
Classification Error E_i	$\frac{f}{n}*100$

the related data point. Classification Error is denoted with E_i, where f is the number of incorrectly classified instances and n denotes the total number of instances in prediction.

5.3 Experimental conditions

Besides novelEnsemble, the results produced from three ensemble learning methods are discussed throughout the paper. This section explains how do four methods exploit experimental data and which configurations are optimal.

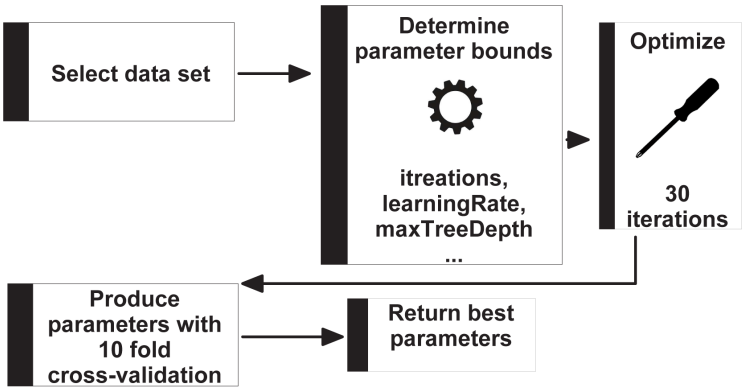

Figure 5. The schema of parameter search.

Parameter search is applied on three methods: novelEnsemble, ensembleHyper, ve triTraining. Figure 5 presents a five-level schema. First level selects the data set to be involved in HO. In the second level, parameter bounds are determined. Parameters of second level are similar for three methods. Notwithstanding this similarity, type and the number of parameter change in accordance with the type of the classifier. In order to find suitable parameters, parameter search methods of third level are exposed to a process including 30 iterations. The values obtained from 10 fold cross-validation are returned in the fifth level.

Firstly, ensembleNoHyper is an ensemble learning method and it does not include any HO process. ensembleNoHyper is run by combining five classifiers with bagging approach. Five classifiers are: RegressionAdaBoostLearner, RegressionRandomForestLearner, RegressionSquareLossGradientBoostLearner, C5.0, and neural net. Configurations of RegressionAdaBoostLearner are Iterations=50,learning rate=1,minSplitSize=1, seed=42, maxTreeDepth=15. minSplitSize represents the value in a node of tree model that must exist before a split is performed. Learning rate affects convergent ratio to local minimal. maxTreeDepth limits depth of a tree model with the predefined value. subSampleRatio is used to overcome class imbalance problem when there is a good model to be trained. featuresPrSplit is the number of features to split in each node. Parameter settings of RegressionAdaBoostLearner are the same that of C5.0. Neural net uses only learningrate=0.1. The parameter settings of RandomForestLearner are trees=100,minSplit=1,maxtreeDepth=2000,
featuresPrSplit=0,subSampleRatio=1,seed=42. RegressionSquareLossGradientBoostLearner employs iterations: 80, learningRate: 0.028, maximumTreeDepth: 12, subSampleRatio: 0.559, featuresPrSplit: 10. In ensembleNoHyper, any parameter search method is not used. Instead, it is exposed to a process including 200 iterations with bagging. In the meantime, a data set is divided into 70% training and 30% testing parts. Mean values of testing results of the prediction and error rates are recorded afterwards.

Specific parameter settings are used for each data set in ensemble-

Hyper. In order to find the optimal settings, firstly parameter search methods are determined. These are RandomSearch and GridSearch. RandomSearch [60] examines random combinations of a range of values of parameters. In the beginning, the number of iterations must be defined. Despite RandomSearch does not gurantee to find the best parameter settings, it is very fast in testing. On the other hand, GridSearch [61] aims to configure optimal parameters for a given model. The main advantage of GridSearch is that it is not limited to be applied to one model. Rather, GridSearch can be utilized for multi-model machine learning parameter search. Average values obtained from the search methods are applied for each data set in accordance with detected settings. Parameter search methods of RegressionAdaBoostLearner are harnessed in the range presented in Table 4. After determining suitable classification settings with 10 fold cross-validation, a process including 200 iterations is performed as in ensembleNoHyper. Average results are obtained with the same training and testing ratios.

In the experiment, triTraining includes RegressionRandomForestLearner, neural net, and C5.0. Some methods such as C5.0 and neural net are involved in the experiment due to their advantages in constructing ensemble methods [8]. To search a parameter, GridSearch has been used. As in preceding methods, triTraining is harnessed to produce average results obtained with 200 iterations.

novelEnsemble is an ensemble learning method that has not restriction on the number of classifiers. To construct novelEnsemble, three classifiers have been used. However, the number of the classifiers constructing novelEnsemble can be augmented. Initially, novelEnsemble selects training data via *BootStrap* to assign them to the related classifier. The ranges presented in Table 4 are of parameters of classifiers given in Algorithm 1. RandomSearch and GridSearch are utilized in searching parameters. To this end, test process including 200 iterations is performed after completing 10 fold cross-validation. Thereafter, performance values are recorded. The distinctive property of novelEnsemble is that it considers classification error along with the number of line of codes and the number of defects.

Table 4. Parameter bounds of experimental classifiers.

RegressionAdaBoostLearner and C5.0				
	Min	**Max**	**Transform**	**Type**
iterations	1	1000		Linear
learningrate	0	0.9		Linear
maxTreeDepth	1	15		Linear
minSplitSize	1	4		Linear
RegressionRandomForestLearner				
	Min	**Max**	**Transform**	**Type**
iterations	1	1000		Linear
splitsize	1	4		Linear
maxTreeDepth	1	15		Linear
RegressionSquareLossGradientBoostLearner				
	Min	**Max**	**Transform**	**Type**
iterations	1	1000		Linear
learningrate	0	1		Linear
maxTreeDepth	1	15		Linear
NeuralNet				
	Min	**Max**	**Transform**	**Type**
iterations	1	1000		Linear
learningrate	0	1		Linear

5.4 Reserch Questions

RQ1: How does novelEnsemble perform comparing with the alternatives in terms of classification error? Iteration number in ensemble learning approaches remarkably affects the success of the prediction. However, error rates are expected to be low. The main purpose of RQ1 is to compare error rates of novelEnsemble with other ensemble approaches. The comparison is done with using novelEnsemble in which parameters are optimized before executing classification process. In doing so, the reliability of ensemble learning approaches can be discussed depending on the testing iteration.

RQ2: Does HO create a favorable affect on ensemble learning meth-

ods? Despite the fact that HO has been investigated on various predictors, RQ2 emerges because any HO technique has not been tried on an ensemble learning method yet. RQ2 compares ensemble learning methods having optimized hyperparameters with the others which have no tuning process. In order to perform this, some performance parameters such as AUC and MCC are utilized.

RQ3: Has the experiment produced different results in terms of statistical values? RQ3 investigates whether the difference of experimental findings are remarkable. To this end, initially Welch's Heteroscedastic F Test is applied on AUC and MCC results. Thereafter, to validate the difference of p-value, Cliff's delta is performed on the performance results.

6 Results

6.1 Classification errors

Answer to RQ1: In order to analyze the relationship between competitive methods, four methods are considered: ensembleHyper, ensembleNoHyper, triTraining and novelEnsemble. Error performance values of three data sets including ant-1.7, camel-1.4 [62], and jedit-4.0 are illustrated in Figure 6-8. Mean error values change according to the iteration of testing process. Three data sets have different project structure and scales. For this reason, they are involved in the error observation so that the generality of the empirical analysis is reinforced. Favorable effects of HO have been detected, especially in novelEnsemble. The case is so different in the others. Unlike novelEnsemble, in which ensemble methods and hyperparameters are different, error values of the others are dramatically high. ensembleHyper has also a HO process that its error rates are noticeably low comparing with ensembleNoHyper. However these two methods are stagnant in three data sets. Thus, fluctuations of error rates are not much depending on the iteration. On the other hand, in such experiments, a linear improvement in error values is inherent. Although triTraining shows a linear improvement, it has the highest error rates.

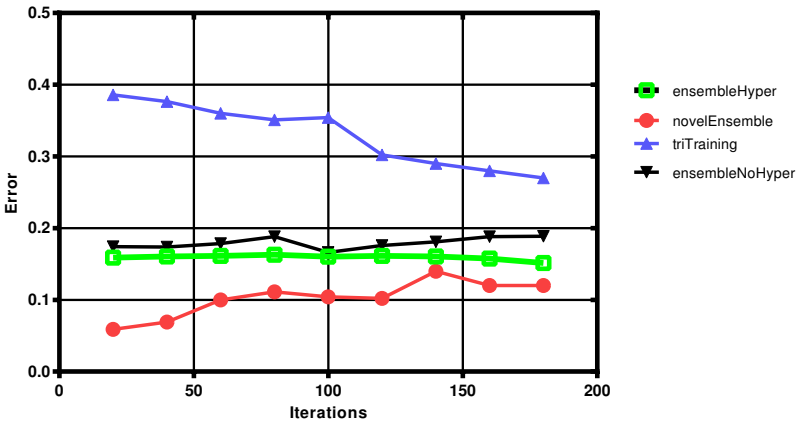

Figure 6. Error rates of ensemble classifiers on ant-1.7.

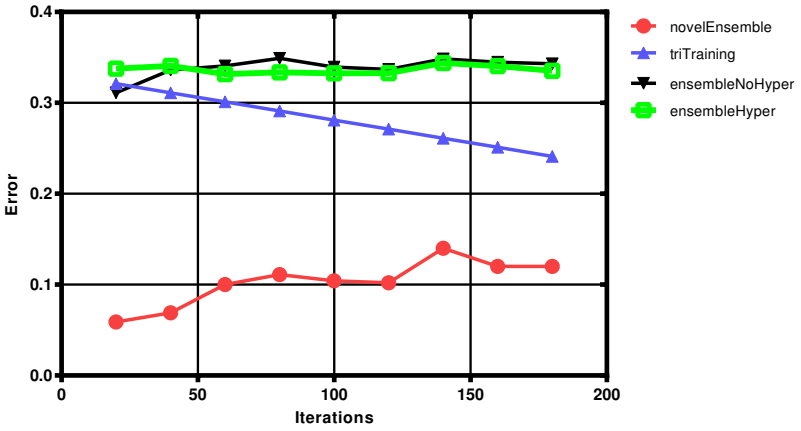

Figure 7. Error rates of ensemble classifiers on camel-1.4.

It is clear that novelEnsemble behaves not like commonly reported classifiers in low iterations. For instance, in Figure 8, the success of novelEnsemble falls especially between 0-50 iterations. However, novelEnsemble seems to be the most preferable method. It is worthwhile to note that optimization is an extremely depended on constraints. Having the number of instances which is not greater than 1000 and limiting the number of iterations with 180 are shortcomings which seriously affect the reliability of the findings.

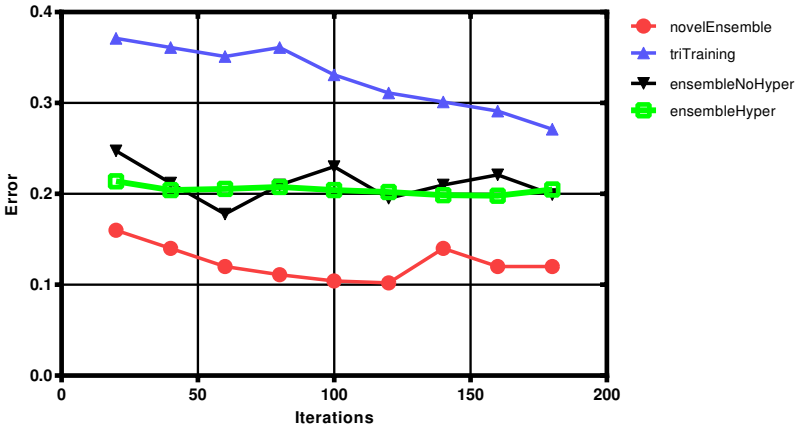

Figure 8. Error rates of ensemble classifiers on jedit-4.0.

6.2 Effects of HO on ensemble learners

Answer to RQ2: novelEnsemble is used along with four methods to answer RQ2. In order to investigate advantages and disadvantages of HO, some methods having HO or not should be added to the experiment. Therefore, HO has not been applied on ensembleNoHyper. Mean values of AUC and MCC results are presented in Figure 9-10. The most scatted values are of ensembleNoHyper that has not any HO process.

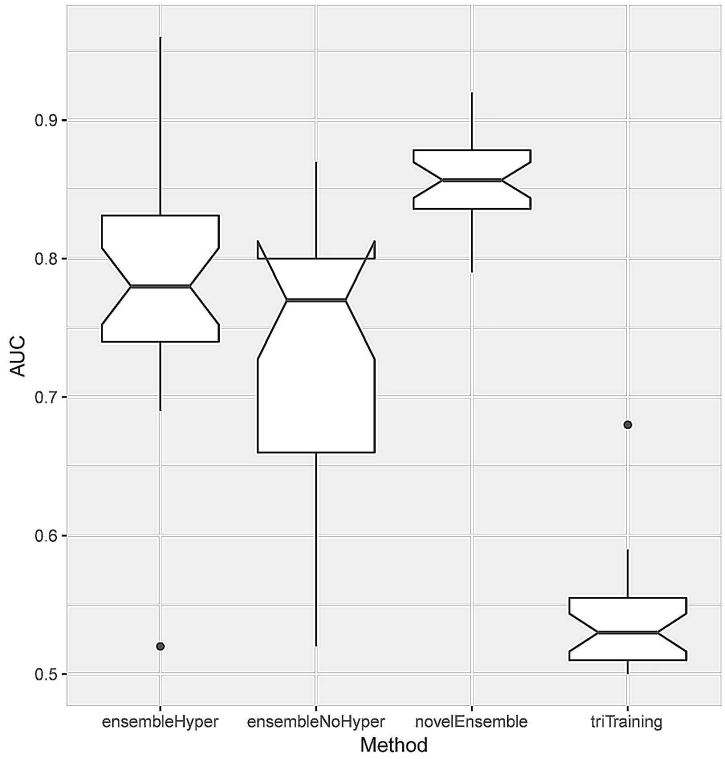

Figure 9. Mean AUC results of four algorithms on all data sets.

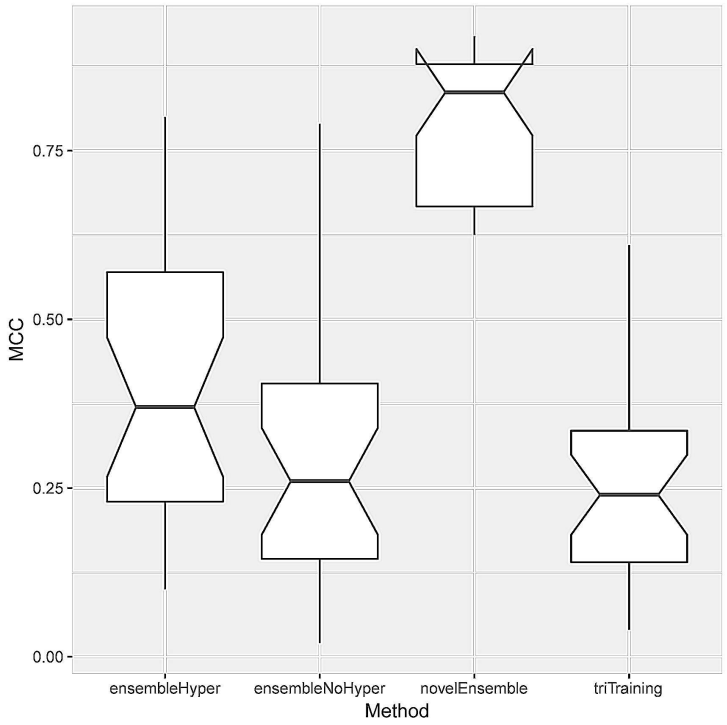

Figure 10. Mean MCC results of four algorithms on all data sets.

It is clearly known from these figures that HO enforces the values to be in a specific interval. In the meantime, triTraining has the lowest mean of AUC values. It has been devised as compatible as working with error values of three classifiers. However, triTraining has not a special evaluation for some classifiers such as C5.0 and neural network which have produced promising results in recent years. Despite the fact that triTraining may produce unfavorable predictions, it is notable that triTraining is a pioneer learner in its field to encourage practitioners to improve future versions of it. Making modifications on the experimental design could achieve a general bias. The figures of the results show that novelEnsemble outperformed its alternatives in terms of the consistency of performance measures. On the other hand, triTaining has even yielded worse results than ensembleNoHyper that does not include any HO process. Although the performance of an ensemble learner seems to be depended on the selected classifiers and data sets, it is also of great importance to parameter search methods. In conclusion, RandomSearch and GridSearch are available in ensembleNoHyper but triTraining does not need such an operation. Further, it has been detected that ensemble learning methods developed based on classification errors such as triTraining cannot remedy the challenges encountered in constructing an ensemble learning method.

Detail AUC results given in Figure 11-14 help to examine the success of data sets individually. In these figures, data sets having high values vary in accordance with their properties. The values recorded in $0.5 - 0.6$ are of triTaining and ensembeNoHyper. ensembleNovelHyper has an AUC value greater than 0.8. Compared to ensembleHyper, triTraining, and ensembleNoHyper, it is clearly seen that HO has increased AUC values at 0.2.

6.3 Statistical analyzes

Answer to RQ3: The results produced by ensemble learning methods differ according to iteration counts. Therefore, evaluating comparison methods for one aspect is difficult and misleading [63]–[65]. To solve this problem, the difference of AUC and MCC results is investigated.

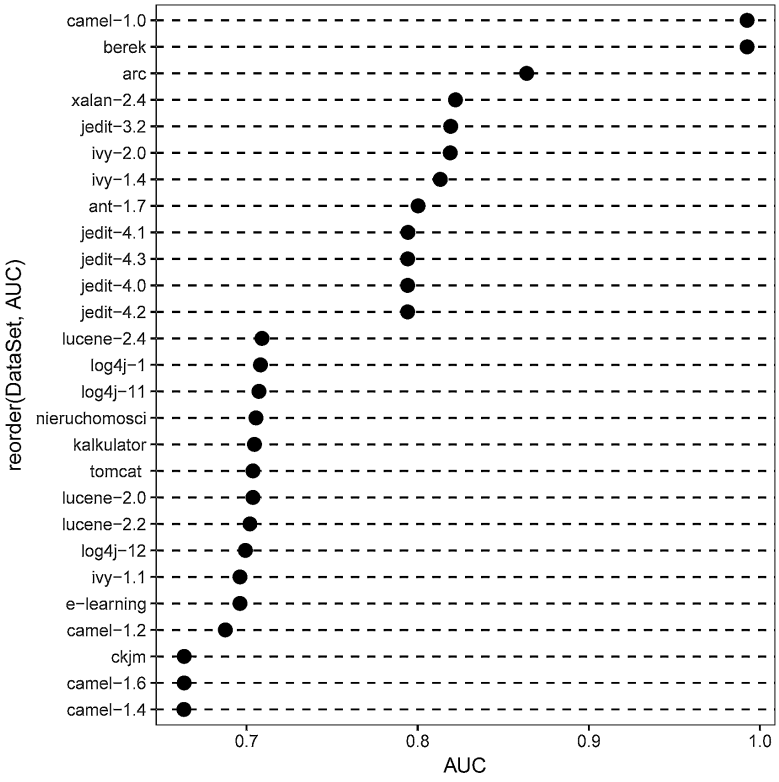

Figure 11. AUC results of ensembleHyper on all data sets.

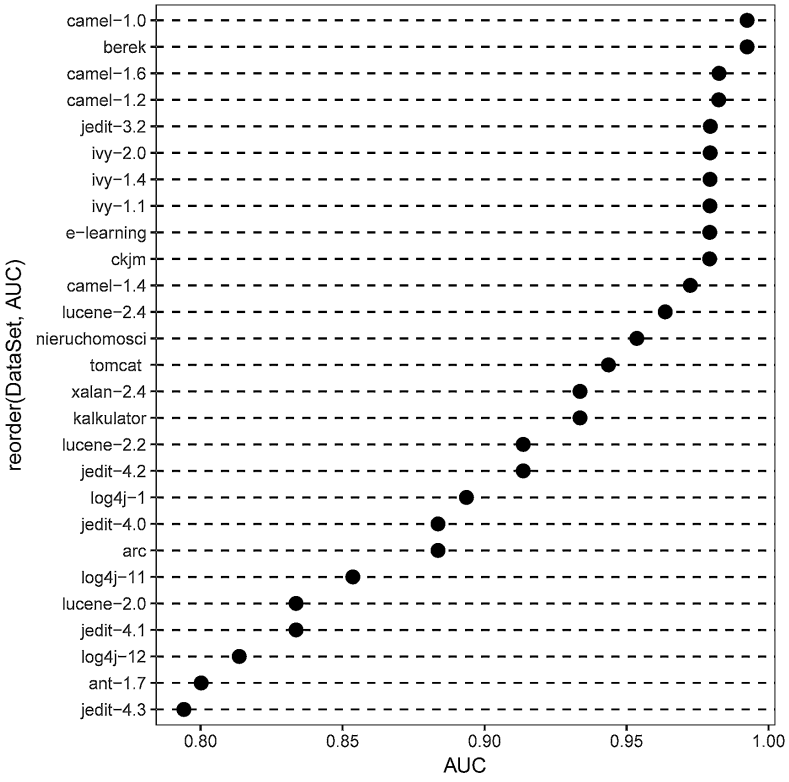

Figure 12. AUC results of novelEnsemble on all data sets.

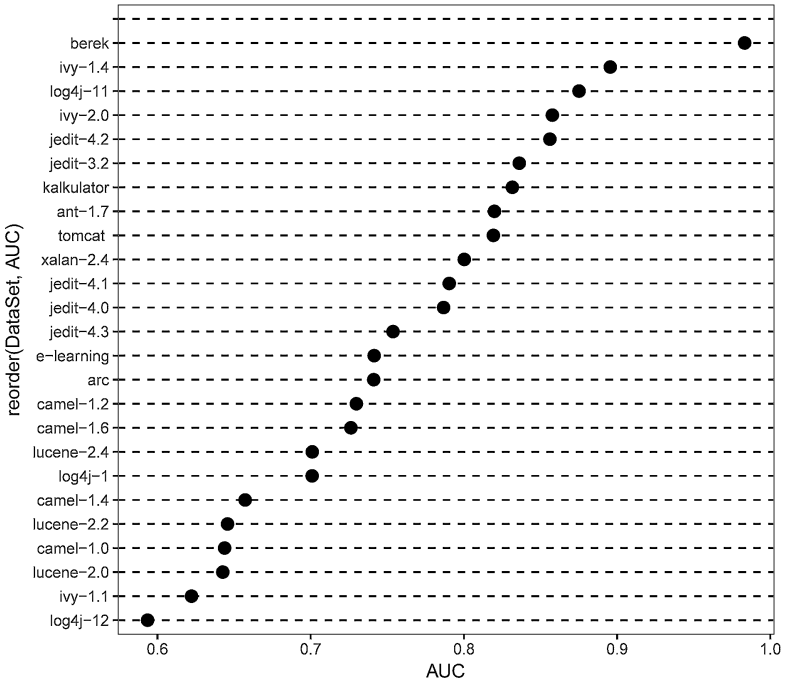

Figure 13. AUC results of ensembleNoHyper on all data sets.

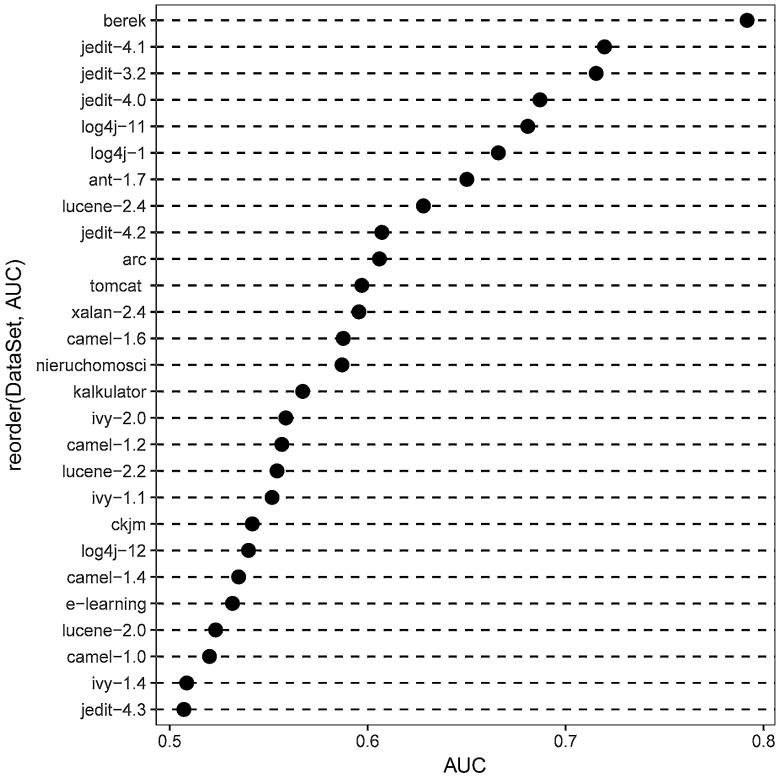

Figure 14. AUC results of triTraining on all data sets.

The data sets exposed to prediction have completely normal distribution thanks to a normalization step of the experiment. However, unequal variances exist in the values of data sets. Welch's Heteroscedastic F Test is suitable in case of unequal variances [66]. It has been applied on AUC and MCC values produced by comparison algorithms. As seen from Table 5, statistical findings are so different in two performance parameters.

Cliff's Delta is selected during the examination of the difference magnitude of performance parameters. The reason is that the results have not any normal distribution. Thus, Cohen's d, which is an alternative method to measure the magnitude of statistical difference, is vulnerable to normality violation. Table 6 presents Cliff's Delta results recorded by benefiting "effsize" library available in R package. The magnitude of the difference is almost remarkable in overall matched methods. In particular, contrary to their structural similarity, delta estimate of novelEnsemble-triTraining is very large. However, the difference is negligible in ensembleHyper-ensembleNoHyper. The types and counts of hyperparameters may have led to the difference. Further, the compatibility level of default configurations could be incidentally high.

Table 5. Welch's Heteroscedastic F Test of performance parameters. Differences are statistically significant for two parameters.

	MCC	**AUC**
statistic	111.308	341.4222
num df	3	3
denom df	56.25705	53.12816
p.value	1.23E-23	1.10E-34

7 Threats to Validity

This section discusses threats to validity for the experiment. Threats are related to data sets, theoretical background of the method, and the

Table 6. Cliff's Delta results of comparison algorithms.

MCC			
	delta estimate	**confidence interval**	**inf sup**
ensembleHyper-novelEnsemble	-0.8024691 (large)	95 percent	-1.4746337
ensembleHyper-ensembleNoHyper	0.1234568 (negligible)	95 percent	-0.1944507 , 0.4178945
novelEnsemble-triTraining	1 (large)	95 percent	0.9976115, 1.0000000
ensembleNoHyper-triTraining	0.877915 (large)	95 percent	0.6848872, 0.9558020
ensembleNoHyper-novelEnsemble	-0.8125685 (large)	95 percent	-1.4746337
ensembleNoHyper-triTraining	0.88294 (large)	95 percent	0.5858982, 0.8534020
AUC			
ensembleHyper-novelEnsemble	-0.5788752 (large)	95 percent	-1.0305861
ensembleHyper-triTraining	0.9122085 (large)	95 percent	0.6950655, 0.9768375
ensembleHyper-ensembleNoHyper	0.1234568 (negligible)	95 percent	-0.1944507 0.4178945
novelEnsemble-triTraining	1 (large)	95 percent	0.9976115, 1.0000000
ensembleNoHyper-triTraining	0.877915 (large)	95 percent	0.6848872, 0.9558020
ensembleNoHyper-novelEnsemble	-0.8024691 (large)	95 percent	-1.4746337

generality of the results.

Threats to internal validity refers to the features used in the experiment. Static code attributes of 27 data sets have been used in the experiment. Therefore, with respect to the ensemble learning, novelEnsemble has a mathematical model based on static code metrics. Having homogeneous metrics has alleviated the burden of experimental process. Similar number of the instances exists in different versions of the projects. It has been tried to select the versions employed in preceding works. The heterogeneity of metrics is one of the main challenges in defect prediction. Since the experiment uses the data sets having same metrics, there is not any need for performing metric matching. However, a replication of this study on heterogeneous software projects would be desirable.

Threats to external validity relates to the data sets which entirely consist of open-source ones. Thus, any industrial data set has not been considered during the experiment. To remove this threat, various versions of data sets of a software project are analyzed.

Threats to construct validity is related to the classifiers utilized in constructing ensemble learners and performance measures. Choosing the classifiers used in the construction of an ensemble learning method is of great importance. Therefore, establishing an ensemble learner via unsuccessful prediction models is not reliable so that superior predictors

such as C5.0 and neural network have been preferred. If classification error is considered, examining error trends in small-scale could create an evaluation threat. To solve this problem, 200 iterations have been performed in testing process. To evaluate the results, AUC, classification error, and MCC have been used. Besides them, g-mean and f-measure are well known among practitioners who work with imbalanced data sets. For this reason, a small proportion of those is not involved in recording performance values.

Threats to conclusion validity refers to the treatment and the outcome. To evaluate the consistency of the results, Welch's Heteroscedastic F Test and Cliff's Delta have been conducted. While choosing statistical methods, it has been regarded whether the data have normal distribution.

8 Conclusion and Future Remarks

HO is a common performance improvement method in machine learning. In this work, the effects of applying HO on ensemble learners have been observed. Further, a novel ensemble learner namely novelEnsemble has been proposed. In preceding works, it has not been investigated how HO is applied on ensemble learning methods. Further, the success of an ensemble learner employing default configurations in defect prediction was not comprehensively examined. In this context, the findings of this study have been obtained from an experimental design solving some issues of defect prediction when ensemble learners are used. novelEnsemble outperformed the others in terms of AUC and MCC. The main findings of the paper can be summarized as follows:

1. novelEnsemble demonstrated a clear superiority to its closest alternative in AUC with 86%. The difference is almost 9%. It rises to 30% when novelEnsemble is compared with triTraining.

2. Some classifiers such as C5.0 and neural network are much more variable than previously thought. Thus, ensembleHyper, which includes some predictors except C5.0 and neural network, shows worse performance than novelEnsemble. Therefore, predictors used in constructing an ensemble learner should be changed according to data sets

and experimental settings.

3. The success of an ensemble learner may not change depending on parameter search method, used ensemble approach, and selected hyperparameters. For instance, ensembleHyper and ensembleNoHyper produced similar results in AUC and MCC values. Moreover, the difference of these values is negligible in performed statistical analyzes. In such cases, the experiment could be replicated by changing hyperparameters and the predictors constructing ensemble learners. The most ineffective configurations can be revealed by that way.

4. In an ensemble learner, besides classification error, other metrics indicating defectiveness should be considered. Thus, novelEnsemble decides the label of an instance by comparing average number of defects with classification error. triTraining has been found to be the most unsuitable ensemble learner among all methods. Because, it has an experimental design based solely on classification error and restricts the number of predictors with three.

The future agenda of the paper encompasses the following: 1) novelEnsemble used *BootStrap* to select training instances as in triTraining. Instead, it is planned to develop a method utilizing some supervised techniques such as clustering by taking data scale into account; 2) Although novelEnsemble is not dependent on the number of classifiers, C5.0 and neural network have been involved in the experiment because they yielded promising results in preceding works. However, it is still unclear to what extent parameter search can contribute to HO in ensemble learners. It would be interesting to investigate that topic; 3) In the experiment, 27 data sets have been used. Yet another direction is to validate the findings of the paper with industrial data sets.

References

[1] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto, "An empirical comparison of model validation techniques for defect prediction models," *IEEE Transactions on Software Engineering*, vol. 43, no. 1, pp. 1–18, 2017.

[2] X. Jing, F. Wu, X. Dong, F. Qi, and B. Xu, "Heterogeneous cross-company defect prediction by unified metric representation and cca-based transfer learning," in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*. ACM, 2015, pp. 496–507.

[3] Y. Zhang, D. Lo, X. Xia, and J. Sun, "An empirical study of classifier combination for cross-project defect prediction," in *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual*, vol. 2. IEEE, 2015, pp. 264–269.

[4] M. Tan, L. Tan, S. Dara, and C. Mayeux, "Online defect prediction for imbalanced data," in *Software Engineering (ICSE), 2015 IEEE/ACM 37th IEEE International Conference on*, vol. 2. IEEE, 2015, pp. 99–108.

[5] X. Yang, K. Tang, and X. Yao, "A learning-to-rank approach to software defect prediction," *IEEE Transactions on Reliability*, vol. 64, no. 1, pp. 234–246, 2015.

[6] T. Wang, Z. Zhang, X. Jing, and L. Zhang, "Multiple kernel ensemble learning for software defect prediction," *Automated Software Engineering*, vol. 23, no. 4, pp. 569–590, 2016.

[7] H. Osman, M. Ghafari, and O. Nierstrasz, "Hyperparameter optimization to improve bug prediction accuracy," in *Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE), IEEE Workshop on*. IEEE, 2017, pp. 33–38.

[8] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto, "Automated parameter optimization of classification techniques for defect prediction models," in *Software Engineering (ICSE), 2016 IEEE/ACM 38th International Conference on*. IEEE, 2016, pp. 321–332.

[9] W. Fu, V. Nair, and T. Menzies, "Why is differential evolution better than grid search for tuning defect predictors?" *arXiv preprint arXiv:1609.02613*, 2016.

[10] W. Fu, T. Menzies, and X. Shen, "Tuning for software analytics: Is it really necessary?" *Information and Software Technology*, vol. 76, pp. 135–146, 2016.

[11] M. Bieshaar, S. Zernetsch, A. Hubert, B. Sick, and K. Doll, "Cooperative starting movement detection of cyclists using convolutional neural networks and a boosted stacking ensemble," *arXiv preprint arXiv:1803.03487*, 2018.

[12] F. Moretti, S. Pizzuti, S. Panzieri, and M. Annunziato, "Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling," *Neurocomputing*, vol. 167, pp. 3–7, 2015.

[13] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.

[14] D. Roschewitz, K. Driessens, and P. Collins, "Simultaneous ensemble generation and hyperparameter optimization for regression," in *Benelux Conference on Artificial Intelligence*. Springer, 2017, pp. 116–130.

[15] X. Wang, H.-J. Xing, Y. Li, Q. Hua, C.-R. Dong, and W. Pedrycz, "A study on relationship between generalization abilities and fuzziness of base classifiers in ensemble learning." *IEEE Trans. Fuzzy Systems*, vol. 23, no. 5, pp. 1638–1654, 2015.

[16] G. Haixiang, L. Yijing, L. Yanan, L. Xiao, and L. Jinling, "Bpso-adaboost-knn ensemble learning algorithm for multi-class imbalanced data classification," *Engineering Applications of Artificial Intelligence*, vol. 49, pp. 176–193, 2016.

[17] M.-J. Kim, D.-K. Kang, and H. B. Kim, "Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1074–1082, 2015.

[18] A. Reiss, G. Hendeby, and D. Stricker, "A novel confidence-based multiclass boosting algorithm for mobile physical activity monitoring," *Personal and Ubiquitous Computing*, vol. 19, no. 1, pp. 105–121, 2015.

[19] Q. Miao, Y. Cao, G. Xia, M. Gong, J. Liu, and J. Song, "Rboost: label noise-robust boosting algorithm based on a nonconvex loss function and the numerically stable base learners," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 11, pp. 2216–2228, 2016.

[20] D. Ryu, O. Choi, and J. Baik, "Value-cognitive boosting with a support vector machine for cross-project defect prediction," *Empirical Software Engineering*, vol. 21, no. 1, pp. 43–71, 2016.

[21] T. Zimmermann, N. Nagappan, H. Gall, E. Giger, and B. Murphy, "Cross-project defect prediction: a large scale experiment on data vs. domain vs. process," in *Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*. ACM, 2009, pp. 91–100.

[22] M. M. Öztürk, "Comparing hyperparameter optimization in cross- and within-project defect prediction: A case study," *Arabian Journal for Science and Engineering*, pp. 1–16, 2018.

[23] Y. Bengio, "Gradient-based optimization of hyperparameters," *Neural computation*, vol. 12, no. 8, pp. 1889–1900, 2000.

[24] S. S. Keerthi, "Efficient tuning of svm hyperparameters using radius/margin bound and iterative algorithms," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1225–1229, 2002.

[25] K. Ito and R. Nakano, "Optimizing support vector regression hyperparameters based on cross-validation," in *Neural Networks, 2003. Proceedings of the International Joint Conference on*, vol. 3. IEEE, 2003, pp. 2077–2082.

[26] I. H. Laradji, M. Alshayeb, and L. Ghouti, "Software defect prediction using ensemble learning on selected features," *Information and Software Technology*, vol. 58, pp. 388–402, 2015.

[27] F. N. Koutanaei, H. Sajedi, and M. Khanbabaei, "A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring," *Journal of Retailing and Consumer Services*, vol. 27, pp. 11–23, 2015.

[28] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Transactions on knowledge and Data Engineering*, vol. 17, no. 11, pp. 1529–1541, 2005.

[29] C. W. Yohannese, T. Li, M. Simfukwe, and F. Khurshid, "Ensembles based combined learning for improved software fault prediction: A comparative study," in *Intelligent Systems and Knowledge Engineering (ISKE), 2017 12th International Conference on*. IEEE, 2017, pp. 1–6.

[30] Z. Sun, Q. Song, and X. Zhu, "Using coding-based ensemble learning to improve software defect prediction," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1806–1817, 2012.

[31] S. Wang and X. Yao, "Using class imbalance learning for software defect prediction," *IEEE Transactions on Reliability*, vol. 62, no. 2, pp. 434–443, 2013.

[32] X. Yang, D. Lo, X. Xia, and J. Sun, "Tlel: A two-layer ensemble learning approach for just-in-time defect prediction," *Information and Software Technology*, vol. 87, pp. 206–220, 2017.

[33] A. Arcuri and G. Fraser, "Parameter tuning or default values? an empirical investigation in search-based software engineering," *Empirical Software Engineering*, vol. 18, no. 3, pp. 594–623, 2013.

[34] A. S. Sayyad, K. Goseva-Popstojanova, T. Menzies, and H. Ammar, "On parameter tuning in search based software engineering:

A replicated empirical study," in *Replication in Empirical Software Engineering Research (RESER), 2013 3rd International Workshop on*. IEEE, 2013, pp. 84–90.

[35] M. Borg, "Tuner: a framework for tuning software engineering tools with hands-on instructions in r," *Journal of software: Evolution and Process*, vol. 28, no. 6, pp. 427–459, 2016.

[36] H. Lappalainen, "Ensemble learning for independent component analysis," in *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA99)*. Citeseer, 1999, pp. 7–12.

[37] Y. Liu and X. Yao, "A cooperative ensemble learning system," in *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on*, vol. 3. IEEE, 1998, pp. 2202–2207.

[38] G. Rätsch, B. Schölkopf, A. J. Smola, S. Mika, T. Onoda, and K.-R. Müller, "Robust ensemble learning," 2000.

[39] L. Guo, Y. Ma, B. Cukic, and H. Singh, "Robust prediction of fault-proneness by random forests," in *Software Reliability Engineering, 2004. ISSRE 2004. 15th International Symposium on*. IEEE, 2004, pp. 417–428.

[40] B. Clark and D. Zubrow, "How good is the software: a review of defect prediction techniques," in *Software Engineering Symposium, Carreige Mellon University*, 2001.

[41] A. Tosun, B. Turhan, and A. Bener, "Ensemble of software defect predictors: a case study," in *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*. ACM, 2008, pp. 318–320.

[42] B. E. Rosen, "Ensemble learning using decorrelated neural networks," *Connection science*, vol. 8, no. 3-4, pp. 373–384, 1996.

[43] D. Barber and C. M. Bishop, "Ensemble learning for multi-layer networks," in *Advances in neural information processing systems*, 1998, pp. 395–401.

[44] R. Polikar, L. Upda, S. S. Upda, and V. Honavar, "Learn++: An incremental learning algorithm for supervised neural networks," *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, vol. 31, no. 4, pp. 497–508, 2001.

[45] N. Liu and H. Wang, "Ensemble based extreme learning machine," *IEEE Signal Processing Letters*, vol. 17, no. 7, p. 754, 2010.

[46] G. I. Webb and Z. Zheng, "Multistrategy ensemble learning: Reducing error by combining ensemble learning techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 8, pp. 980–991, 2004.

[47] P. Zhang, X. Zhu, Y. Shi, L. Guo, and X. Wu, "Robust ensemble learning for mining noisy data streams," *Decision Support Systems*, vol. 50, no. 2, pp. 469–479, 2011.

[48] J. Hu, T. Li, C. Luo, H. Fujita, and Y. Yang, "Incremental fuzzy cluster ensemble learning based on rough set theory," *Knowledge-Based Systems*, vol. 132, pp. 144–155, 2017.

[49] E. I. Papageorgiou and A. Kannappan, "Fuzzy cognitive map ensemble learning paradigm to solve classification problems: Application to autism identification," *Applied Soft Computing*, vol. 12, no. 12, pp. 3798–3809, 2012.

[50] M. Pratama, W. Pedrycz, and E. Lughofer, "Evolving ensemble fuzzy classifier," *IEEE Transactions on Fuzzy Systems*, 2018.

[51] Y. Peng, G. Kou, G. Wang, W. Wu, and Y. Shi, "Ensemble of software defect predictors: an ahp-based evaluation method," *International Journal of Information Technology & Decision Making*, vol. 10, no. 01, pp. 187–206, 2011.

[52] U. Apache, "Apache software foundation," *URL http://java. apache. org*, 2011.

[53] D. Spinellis, "Ckjm-a tool for calculating chidamber and kemerer java metrics," 2009.

[54] M. Jureczko and L. Madeyski, "Towards identifying software project clusters with regard to defect prediction," in *Proceedings of the 6th International Conference on Predictive Models in Software Engineering*. ACM, 2010, p. 9.

[55] L. Chen, B. Fang, Z. Shang, and Y. Tang, "Tackling class overlap and imbalance problems in software defect prediction," *Software Quality Journal*, vol. 26, no. 1, pp. 97–125, 2018.

[56] X.-Y. Jing, F. Wu, X. Dong, and B. Xu, "An improved sda based defect prediction framework for both within-project and cross-project class-imbalance problems," *IEEE Transactions on Software Engineering*, vol. 43, no. 4, pp. 321–339, 2017.

[57] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto, "The impact of automated parameter optimization on defect prediction models," *IEEE Transactions on Software Engineering*, 2018.

[58] M. Shepperd, D. Bowes, and T. Hall, "Researcher bias: The use of machine learning in software defect prediction," *IEEE Transactions on Software Engineering*, vol. 40, no. 6, pp. 603–616, 2014.

[59] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[60] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.

[61] R. Everaers and K. Kremer, "A fast grid search algorithm for molecular dynamics simulations with short-range interactions,"

Computer Physics Communications, vol. 81, no. 1-2, pp. 19–55, 1994.

[62] C. Ibsen and J. Anstey, *Camel in action*. Manning Publications Co., 2018.

[63] A. Kotelyanskii and G. M. Kapfhammer, "Parameter tuning for search-based test-data generation revisited: Support for previous results," in *Quality Software (QSIC), 2014 14th International Conference on*. IEEE, 2014, pp. 79–84.

[64] S. Herbold, A. Trautsch, and J. Grabowski, "A comparative study to benchmark cross-project defect prediction approaches," *IEEE Transactions on Software Engineering*, 2017.

[65] F. Wu, X.-Y. Jing, X. Dong, J. Cao, M. Xu, H. Zhang, S. Ying, and B. Xu, "Cross-project and within-project semi-supervised software defect prediction problems study using a unified solution," in *Software Engineering Companion (ICSE-C), 2017 IEEE/ACM 39th International Conference on*. IEEE, 2017, pp. 195–197.

[66] S.-L. Jan and G. Shieh, "Sample size determinations for welch's test in one-way heteroscedastic anova," *British Journal of Mathematical and Statistical Psychology*, vol. 67, no. 1, pp. 72–93, 2014.

M. Maruf Öztürk

Received December 7, 2018

Department of
Computer Engineering
Faculty of Engineering
Isparta, TURKEY

Phone: +90 246 211 15 63
E–mail: muhammedozturk@sdu.edu.tr, maruf215@gmail.com