ISSN 2579-2784 (Print)
ISSN 2538-2788 (Online)

MATHEMATICAL PROBLEMS OF COMPUTER SCIENCE

LVIII

Yerevan
2022

Հայաստանի Հանրապետության Գիտությունների ազգային ակադեմիայի Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ

Институт проблем информатики и автоматизации Национальной академии наук Республики Армения

Institute for Informatics and Automation Problems of the National Academy of Sciences of the Republic of Armenia

Կոմպյուտերային գիտության մաթեմատիկական խնդիրներ

Математические проблемы компьютерных наук

Mathematical Problems of Computer Science

LVIII

ՀՐԱՏԱՐԱԿՎԱԾ Է ՀՀ ԳԱԱ ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԵՎ ԱՎՏՈՄԱՏԱՑՄԱՆ ՊՐՈԲԼԵՄՆԵՐԻ ԻՆՍՏԻՏՈՒՏԻ ԿՈՂՄԻՑ
ОПУБЛИКОВАНО ИНСТИТУТОМ ПРОБЛЕМ ИНФОРМАТИКИ И АВТОМАТИЗАЦИИ НАН РА
PUBLISHED BY INSTITUTE FOR INFORMATICS AND AUTOMATION PROBLEMS OF NAS RA

ԵՐԵՎԱՆ **2022** **YEREVAN**

Կոմպյուտերային գիտության մաթեմատիկական խնդիրներ, LVIII

Կոմպյուտերային գիտության մաթեմատիկական խնդիրներ պարբերականը հրատարակվում է տարեկան երկու անգամ ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտի (ԻԱՊԻ) կողմից: Այն ընդգրկում է տեսական և կիրառական մաթեմատիկայի, ինֆորմատիկայի և հաշվողական տեխնիկայի ժամանակակից ուղղությունները:

Այն ընդգրկված է Բարձրագույն որակավորման հանձնաժողովի ընդունելի ամսագրերի ցանկում:

Տպագրվում է Խմբագրական խորհրդի 2022թ. նոյեմբերի 25-ի N 22-11/2 նիստի որոշման հիման վրա

ISSN 2579-2784 (Print)

ISSN 2738-2788 (Online)

Журнал **Математические проблемы компьютерных наук** издается два раза в год Институтом проблем информатики и автоматизации НАН РА. Он охватывает современные направления теоретической и прикладной математики, информатики и вычислительной техники.

Он включен в список допустимых журналов Высшей квалификационной комиссии.

Печатается на основании решения N 22-11/2 заседания
Редакционного совета от 25 ноября 2022г.

ISSN 2579-2784 (Print)

ISSN 2738-2788 (Online)

Mathematical Problems of Computer Science, LVIII

The periodical **Mathematical Problems of Computer Science** is published twice per year by the Institute for Informatics and Automation Problems of NAS RA. It covers modern directions of theoretical and applied mathematics, informatics and computer science.

It is included in the list of acceptable journals of the Higher Qualification Committee.

Printed on the basis of decision N 22-11/2 of session of the Editorial Council dated November 25, 2022.

ISSN 2579-2784 (Print)

ISSN 2738-2788 (Online)

CONTENTS

Mathematical Problems of Computer Science 58, 7–19, 2022.
doi:10.51408/1963-0088

UDC 519.6

Analytical Inversion of Tridiagonal Hermitian Matrices

Yuri R. Hakopian and Avetik H. Manukyan

Yerevan State University, Yerevan, Armenia
e-mail: yuri.hakopian@ysu.am, avetiq.manukyan1@ysumail.am

Abstract

In this paper we give an algorithm for inverting complex tridiagonal Hermitian matrices with optimal computational efforts. For matrices of a special form and, in particular, for Toeplitz matrices, the derived formulas lead to closed-form expressions for the elements of inverse matrices.
Keywords: Inverse matrix, Tridiagonal matrix, Hermitian matrix, Toeplitz matrix.
Article info: Received 21 April 2022; received in revised form 15 July 2022; accepted 23 August 2022.

1. Introduction

Tridiagonal matrices are encountered in many areas of applied mathematics. Such matrices are of great importance in finite difference and finite element methods for differential equations. The construction of cubic splines is reduced to solving systems with tridiagonal matrices. Symmetric matrices are reduced to tridiagonal matrices by the similarity Householder transformation (see [1, 2, 3], for instance). Other examples can be cited.

There is a well-known fast numerical method for solving systems with tridiagonal matrices. At the same time, the analytical matrix inversion is also of certain interest (see [4, 5, 6], for instance). For tridiagonal matrices of special types, this leads to closed-form expressions for the elements of inverse matrices [7, 8, 9, 10]. This is undoubtedly useful in theoretical considerations. Further, explicit formulas can be a part of more general computational procedures. There are other reasons as well.

In this article, we focus our attention on complex Hermitian tridiagonal matrices. We will construct a fairly simple computational procedure, consisting of a sequence of recurrence relations, leading to the calculation of the elements of the inverse matrix. In special cases, in particular for Toeplitz tridiagonal Hermitian matrices, the procedure can become the basis for deriving closed-form expressions for the elements of the inverse matrix.

We note right away that throughout this article $\overline{z}$ stands for the complex conjugate of the complex number z.

Let a nonsingular tridiagonal Hermitian matrix

$$A = \begin{bmatrix} a_1 & b_1 & & & \\ \overline{b_1} & a_2 & b_2 & & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & \overline{b_{n-2}} & a_{n-1} & b_{n-1} \\ & & & \overline{b_{n-1}} & a_n \end{bmatrix} \tag{1}$$

be given, where a_i, $i = 1, 2, \ldots, n$ are real numbers and $b_i \neq 0$ for $i = 1, 2, \ldots, n-1$. In accordance with the accepted notation, $A = A^*$. We assume that $n > 3$. The requirement that the subdiagonal (superdiagonal) elements of the matrix be nonzero is not restrictive. Indeed, if some of these elements are equal to zero, the problem of computing the inverse matrix is decomposed into several similar problems for tridiagonal matrices of lower order.

2. Preliminary Calculations

Let $A^{-1} = [x_{ij}]_{n\times n}$. This matrix is also Hermitian. In our considerations we will use the notation

$$X^{(j)} \equiv [x_{1\,j}\, x_{2\,j} \ldots x_{n\,j}]^T, \quad j = 1, 2, \ldots, n$$

for the columns of the inverse matrix.

The matrix A can be represented as a product

$$A = DB \tag{2}$$

of the matrices

$$D = diag\,[b_1, \overline{b_1}, \overline{b_2}, \ldots, \overline{b_{n-2}}, \overline{b_{n-1}}] \tag{3}$$

and

$$B = \begin{bmatrix} p & 1 & & & & \\ 1 & f_2 & g_2 & & 0 & \\ & 1 & f_3 & g_3 & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & 1 & f_{n-1} & g_{n-1} \\ & & & & 1 & q \end{bmatrix}, \tag{4}$$

where

$$f_i = \frac{a_i}{\overline{b_{i-1}}},\; g_i = \frac{b_i}{\overline{b_{i-1}}},\; i = 2, 3, \ldots, n-1;\; p = \frac{a_1}{b_1},\; q = \frac{a_n}{\overline{b_{n-1}}}. \tag{5}$$

Having a nonsingular matrix B defined in (4), let us consider the following system of linear algebraic equations

$$\begin{aligned} &p\mu_1 + \mu_2 = \alpha \\ &\mu_{i-1} + f_i\mu_i + g_i\mu_{i+1} = 0,\;\; 2 \leq i \leq n-1 \\ &\mu_{n-1} + q\mu_n = 0, \end{aligned} \tag{6}$$

where we will set the right-hand side α of the first equation a little later. It is easy to verify that regardless of the choice of α, the recursively defined quantities

$$\begin{aligned} &\mu_n = 1\,,\; \mu_{n-1} = -q\,, \\ &\mu_{i-1} = -f_i\mu_i - g_i\mu_{i+1}\,,\; i = n-1, n-2, \ldots, 2 \end{aligned} \tag{7}$$

satisfy all equations of the system (6), starting with the second one. Then, we choose the quantity α as follows:

$$\alpha = p\mu_1 + \mu_2. \tag{8}$$

Remark 1 *Since, by assumption, the matrix B is nonsingular (it follows from (2)), then $\alpha \neq 0$. Indeed, otherwise we would have obtained that the homogeneous system (6) has a nontrivial solution. Further,*

$$\alpha = \frac{a_1}{b_1}\mu_1 + \mu_2 = \frac{1}{b_1}(a_1\mu_1 + b_1\mu_2).$$

Therefore

$$a_1\mu_1 + b_1\mu_2 \neq 0$$

as well.

Thus,

$$\alpha = b_1^{-1} t^{-1}, \tag{9}$$

where

$$t \equiv (a_1\mu_1 + b_1\mu_2)^{-1}. \tag{10}$$

Let us introduce the vector

$$r^{(1)} \equiv [\mu_1\, \mu_2 \dots \mu_n]^T,$$

the components of which are specified in (7). As follows from (4), (6) and (9),

$$Br^{(1)} = [\alpha\, 0 \dots 0]^T = \alpha e^{(1)} = b_1^{-1} t^{-1} e^{(1)},$$

where $e^{(1)} \equiv [1\, 0 \dots 0]^T$. Further, on the basis of factorization (2) of the matrix A, we obtain the equality

$$Ar^{(1)} = DBr^{(1)} = b_1^{-1} t^{-1} De^{(1)} = t^{-1} e^{(1)}; \tag{11}$$

here we have used the obvious equality $De^{(1)} = b_1 e^{(1)}$ (see (3)). The equality (11) allows to compute the first column of the inverse matrix A^{-1}. Indeed, from this equality we find that

$$A^{-1} e^{(1)} = tr^{(1)}.$$

Since $A^{-1} e^{(1)} = X^{(1)}$, then $X^{(1)} = tr^{(1)}$, or

$$x_{i1} = t\mu_i, \quad i = 1, 2, \dots, n. \tag{12}$$

Thus, we have found the first column of the inverse matrix. Similarly, we can calculate the last column of the matrix A^{-1}. For this purpose, let us consider the linear system

$$\begin{aligned} &p\nu_1 + \nu_2 = 0 \\ &\nu_{i-1} + f_i\nu_i + g_i\nu_{i+1} = 0, \;\; 2 \leq i \leq n-1 \\ &\nu_{n-1} + q\nu_n = \beta, \end{aligned} \tag{13}$$

where we will set the right-hand side β of the last equation later. Regardless of the choice of β, the recursively defined quantities

$$\begin{aligned} &\nu_1 = 1\,,\; \nu_2 = -p\,, \\ &\nu_{i+1} = -\frac{1}{g_i}(\nu_{i-1} + f_i\nu_i)\,,\; i = 2, 3, \dots, n-1 \end{aligned} \tag{14}$$

satisfy the first $n-1$ equations of the system (13). Then we choose the quantity β as follows:

$$\beta = \nu_{n-1} + q\nu_n. \tag{15}$$

Since the matrix B is nonsingular, then $\beta \neq 0$ (see Remark 1). Substituting the expression of the quantity q given in (5) into (15) yields

$$\beta = \nu_{n-1} + \frac{a_n}{\overline{b_{n-1}}}\nu_n = \frac{1}{\overline{b_{n-1}}}(\overline{b_{n-1}}\nu_{n-1} + a_n\nu_n).$$

Thus,

$$\beta = \overline{b_{n-1}}^{\,-1}\theta^{-1}, \tag{16}$$

where

$$\theta \equiv (\overline{b_{n-1}}\nu_{n-1} + a_n\nu_n)^{-1}.$$

Now let us introduce the vector

$$r^{(n)} \equiv [\nu_1\, \nu_2 \ldots \nu_n]^T,$$

the components of which are specified in (14). From (4), (13) and (16) we find that

$$Br^{(n)} = [0, \ldots 0\, \beta]^T = \beta e^{(n)} = \overline{b_{n-1}}^{\,-1}\theta^{-1}e^{(n)},$$

where $e^{(n)} \equiv [0 \ldots 0\, 1]^T$. Having the factorization (2) of the matrix A, we obtain the equality

$$Ar^{(n)} = DBr^{(n)} = \overline{b_{n-1}}^{\,-1}\theta^{-1}De^{(n)} = \theta^{-1}e^{(n)}.$$

From here,

$$A^{-1}e^{(n)} = \theta r^{(n)}.$$

Since $A^{-1}e^{(n)} = X^{(n)}$, then $X^{(n)} = \theta r^{(n)}$, or

$$x_{in} = \theta\nu_i, \quad i = 1, 2, \ldots, n. \tag{17}$$

Let us refine the last expression. From (12), $x_{n1} = t\mu_n = t$. Further, according to (17), $x_{1n} = \theta\nu_1 = \theta$. Since A^{-1} is a Hermitian matrix, then $x_{1n} = \overline{x_{n1}}$. Consequently, $\theta = \bar{t}$, and we come to the conclusion that

$$x_{in} = \bar{t}\nu_i, \quad i = 1, 2, \ldots, n. \tag{18}$$

So, we have found the first and the last columns of the Hermitian matrix A^{-1}. These are expressions (12) and (18). Taking into account that $\nu_1 = 1$ and $\mu_n = 1$, we write these elements in the form of

$$x_{i1} = t\mu_i\overline{\nu_1}, \; x_{in} = \bar{t}\,\overline{\mu_n}\nu_i, \quad i = 1, 2, \ldots, n. \tag{19}$$

Moreover, the diagonal elements $x_{11} = t\mu_1\overline{\nu_1}$ and $x_{nn} = \bar{t}\,\overline{\mu_n}\nu_n$ are real numbers. Therefore, we can write $x_{nn} = t\mu_n\overline{\nu_n}$ as well.

Looking ahead, we say that in the next section we will prove that the quantities

$$t\mu_i\overline{\nu_i}, \quad i = 2, 3, \ldots, n-1 \tag{20}$$

are the remaining diagonal elements of the matrix A^{-1}. To do this, here we first establish that the quantities (20) are real numbers (naturally, without assuming that they are somehow related to the matrix A^{-1}).

Let us introduce into consideration the quantities

$$R_i \equiv b_{i-1}(t\mu_i\overline{\nu_{i-1}}) + \overline{b_{i-1}}(t\mu_{i-1}\overline{\nu_i}), \quad i = 2, 3, \ldots, n-2. \tag{21}$$

Lemma 1. *The quantity R_2 is a real number.*

Proof. Since $\nu_1 = 1$ and $\nu_2 = -p$ (see (2.13)), then

$$R_2 = t(b_1\mu_2\overline{\nu_1} + \overline{b_1}\mu_1\overline{\nu_2}) = tb_1(\mu_2 - p\mu_1).$$

Further, taking into account the equalities (8) and (9), we get

$$R_2 = tb_1(\alpha - 2p\mu_1) = tb_1\alpha - 2pb_1(t\mu_1) = 1 - 2a_1(t\mu_1).$$

The quantities a_1 and $t\mu_1$ are real numbers, so R_2 is also a real number. □

Lemma 2. *The quantities R_i from (21) satisfy the relations*

$$R_i = -R_{i-1} - 2a_{i-1}(t\mu_{i-1}\overline{\nu_{i-1}}), \quad i = 3, 4, \ldots, n-2. \tag{22}$$

Proof. From (6) we have the equality

$$\mu_{i-2} + f_{i-1}\mu_{i-1} + g_{i-1}\mu_i = 0.$$

Using formulas (5), let us write this equality in the form of

$$\overline{b_{i-2}}\mu_{i-2} + a_{i-1}\mu_{i-1} + b_{i-1}\mu_i = 0.$$

Multiplying both parts of the last equality by $t\overline{\nu_{i-1}}$, we get that

$$b_{i-1}(t\mu_i\overline{\nu_{i-1}}) = -\overline{b_{i-2}}(t\mu_{i-2}\overline{\nu_{i-1}}) - a_{i-1}(t\mu_{i-1}\overline{\nu_{i-1}}). \tag{23}$$

Similarly, from (13) we have the equality

$$\nu_{i-2} + f_{i-1}\nu_{i-1} + g_{i-1}\nu_i = 0,$$

which can be written as follows:

$$b_{i-2}\overline{\nu_{i-2}} + a_{i-1}\overline{\nu_{i-1}} + \overline{b_{i-1}}\overline{\nu_i} = 0.$$

Multiplying both parts of this equality by $t\mu_{i-1}$ yields

$$\overline{b_{i-1}}(t\mu_{i-1}\overline{\nu_i}) = -b_{i-2}(t\mu_{i-1}\overline{\nu_{i-2}}) - a_{i-1}(t\mu_{i-1}\overline{\nu_{i-1}}). \tag{24}$$

The relation (22) follows directly from the equalities (23) and (24). □

Lemma 3. *The quantities $t\mu_i\overline{\nu_i}$, $\quad i = 2, 3, \ldots, n-1$ are real numbers.*

Proof. Consider first the quantity $t\mu_2\overline{\nu_2}$. Since $p\mu_1 + \mu_2 = \alpha$ and $\nu_2 = -p$ (see (6) and (14)), then

$$t\mu_2\overline{\nu_2} = t(p\mu_1 - \alpha)\overline{p} = (p\overline{p})(t\mu_1) - t\alpha\overline{p}.$$

Further, using the equality (9), we obtain that

$$t\mu_2\overline{\nu_2} = (p\overline{p})(t\mu_1) - \frac{\overline{p}}{b_1} = (p\overline{p})(t\mu_1) - \frac{a_1}{b_1\overline{b_1}}.$$

Thus, the quantity $t\mu_2\overline{\nu_2}$ is a real number.

Next, consider the quantity $t\mu_3\overline{\nu_3}$. As follows from (6) and (13),

$$\mu_3 = -\frac{a_2}{b_2}\mu_2 - \frac{\overline{b_1}}{b_2}\mu_1, \quad \overline{\nu_3} = -\frac{a_2}{\overline{b_2}}\overline{\nu_2} - \frac{b_1}{\overline{b_2}}\overline{\nu_1}.$$

Proceeding from these equalities, we get that

$$t\mu_3\overline{\nu_3} = \frac{1}{b_2\overline{b_2}}\left[a_2^2(t\mu_2\overline{\nu_2}) + b_1\overline{b_1}(t\mu_1\overline{\nu_1}) + a_2R_2\right].$$

The quantities $t\mu_1\overline{\nu_1}$ and $t\mu_2\overline{\nu_2}$ are real numbers. According to Lemma 1, the quantity R_2 is also a real number. Therefore, $t\mu_3\overline{\nu_3}$ is a real number as well.

Further reasoning will be carried out by the method of mathematical induction on i. Suppose that for some value of i, where $3 \le i \le n-2$, it is already known that the quantities $t\mu_k\overline{\nu_k}$, $k \le i$ and R_k, $k \le i-1$ are real numbers. From (6) and (13) we have

$$\mu_{i+1} = -\frac{a_i}{b_i}\mu_i - \frac{\overline{b_{i-1}}}{b_i}\mu_{i-1}, \quad \overline{\nu_{i+1}} = -\frac{a_i}{\overline{b_i}}\overline{\nu_i} - \frac{b_{i-1}}{\overline{b_i}}\overline{\nu_{i-1}}.$$

Then

$$t\mu_{i+1}\overline{\nu_{i+1}} = \frac{1}{b_i\overline{b_i}}\left[a_i^2(t\mu_i\overline{\nu_i}) + b_{i-1}\overline{b_{i-1}}(t\mu_{i-1}\overline{\nu_{i-1}}) + a_iR_i\right].$$

Hence, by virtue of the assumptions made and taking into account the assertion of Lemma 2, we arrive at a conclusion that the quantity $t\mu_{i+1}\overline{\nu_{i+1}}$ is a real number. $\square$

Remark 2 *We have established that the quantities $t\mu_i\overline{\nu_i}$, $i = 1, 2, \ldots, n$ are real numbers. Therefore, $t\mu_i\overline{\nu_i} = \bar{t}\,\overline{\mu_i}\nu_i$.*

3. The Elements of the Inverse Matrix

Above we obtained the expressions (19) for the elements of the first and the last columns of the inverse matrix, as well as some auxiliary statements. Based on these results, here we derive formulas for the remaining elements of the inverse matrix.

Let $2 \le j \le n-1$. We introduce into consideration the vector

$$r^{(j)} \equiv [\bar{t}\,\overline{\mu_j}\nu_1, \ldots, \bar{t}\,\overline{\mu_j}\nu_{j-1}, t\mu_j\overline{\nu_j}, t\mu_{j+1}\overline{\nu_j}, \ldots, t\mu_n\overline{\nu_j}]^T, \tag{25}$$

where the quantities μ_i and ν_i are specified in (7) and (14), respectively. Multiplying the matrix B defined in (4) and the vector $r^{(j)}$ yields

$$Br^{(j)} = z^{(j)}, \tag{26}$$

where the components of the vector

$$z^{(j)} = [z_1^{(j)}\ z_2^{(j)} \ldots z_{j-1}^{(j)}\ \delta_j\ z_{j+1}^{(j)} \ldots z_{n-1}^{(j)}\ z_n^{(j)}]^T$$

are calculated as follows:

$$\begin{aligned}
z_1^{(j)} &= \bar{t}\,\overline{\mu_j}(p\nu_1 + \nu_2), \\
z_i^{(j)} &= \bar{t}\,\overline{\mu_j}(\nu_{i-1} + f_i\nu_i + g_i\nu_{i+1}), \quad 2 \le i \le j-1, \\
\delta_j &= \bar{t}\,\overline{\mu_j}\nu_{j-1} + f_j(t\mu_j\overline{\nu_j}) + g_j(t\mu_{j+1}\overline{\nu_j}), \\
z_i^{(j)} &= t(\mu_{i-1} + f_i\mu_i + g_i\mu_{i+1})\overline{\nu_j}, \quad j+1 \le i \le n-1, \\
z_n^{(j)} &= t(\mu_{n-1} + q\mu_n)\overline{\nu_j}.
\end{aligned}$$

Having equations (6) and (13), we conclude that $z_i^{(j)} = 0$ for $1 \le i \le j-1$ and $j+1 \le i \le n$. Thus,

$$z^{(j)} = [0\ \dots\ 0\,\delta_j\,0\ \dots\ 0]^T = \delta_j e^{(j)}, \tag{27}$$

where $e^{(j)} = [0\dots0\,1\,0\dots0]^T$ (the unit is located on jth place).

It remains to clarify the quantity δ_j. Taking into account Remark 2, we have

$$\begin{aligned} \delta_j &= \bar{t}\,\overline{\mu_j}\nu_{j-1} + f_j(\bar{t}\,\overline{\mu_j}\nu_j) + g_j(t\mu_{j+1}\overline{\nu_j}) \\ &= \bar{t}\,\overline{\mu_j}(\nu_{j-1} + f_j\nu_j) + g_j(t\mu_{j+1}\overline{\nu_j}). \end{aligned} \tag{28}$$

Since $\nu_{j-1} + f_j\nu_j = -g_j\nu_{j+1}$ (see (13)), then

$$\delta_j = g_j(t\mu_{j+1}\overline{\nu_j} - \bar{t}\,\overline{\mu_j}\nu_{j+1}), \quad 2 \le j \le n-1. \tag{29}$$

Let us get one more representation of the quantity δ_j. Since $g_j\mu_{j+1} = -\mu_{j-1} - f_j\mu_j$ (see (6)), then from(28) it follows that

$$\delta_j = \bar{t}\,\overline{\mu_j}\nu_{j-1} - t\mu_{j-1}\overline{\nu_j} + f_j(\bar{t}\,\overline{\mu_j}\nu_j - t\mu_j\overline{\nu_j}).$$

From here, according to Remark 2, we obtain

$$\delta_j = \bar{t}\,\overline{\mu_j}\nu_{j-1} - t\mu_{j-1}\overline{\nu_j}, \quad 2 \le j \le n-1. \tag{30}$$

Assuming that $3 \le j \le n-1$, we can write the expression (30) in the form of

$$\delta_j = \frac{1}{\overline{g_{j-1}}}\,\overline{g_{j-1}(t\mu_j\overline{\nu_{j-1}} - \bar{t}\,\overline{\mu_{j-1}}\nu_j)}.$$

Comparing with the record (29), we arrive at the relation

$$\delta_j = \frac{1}{\overline{g_{j-1}}}\,\overline{\delta_{j-1}}, \quad 3 \le j \le n-1. \tag{31}$$

Based on the relation (31), one can easily show that

$$\delta_j = \begin{cases} \overline{b_{j-1}}^{-1} b_1 \overline{\delta_2}\,, & \text{if } j \text{ is odd,} \\ \overline{b_{j-1}}^{-1} \overline{b_1} \delta_2\,, & \text{if } j \text{ is even.} \end{cases} \tag{32}$$

Finally, let us calculate the quantity δ_2. According to the representation (30), we have

$$\begin{aligned} \delta_2 &= \bar{t}\,\overline{\mu_2}\nu_1 - t\mu_1\overline{\nu_2} = \bar{t}\,\overline{\mu_2} + t\mu_1\overline{p} \\ &= \bar{t}\,\overline{\mu_2} + \bar{t}\,\overline{\mu_1}\,\overline{p} = \bar{t}\,(\overline{\mu_2} + \overline{p}\,\overline{\mu_1}) = \bar{t}\,\overline{\alpha} = \overline{b_1}^{-1}, \end{aligned} \tag{33}$$

(see (6) and (9)). Thus, from (32) and (33) we conclude that

$$\delta_j = \overline{b_{j-1}}^{-1}, \quad j = 2, 3, \dots, n-1. \tag{34}$$

Summing up the results, from (27) and (34) we come to the equality

$$z^{(j)} = \overline{b_{j-1}}^{-1} e^{(j)}. \tag{35}$$

Proceeding from the factorization (2) of the matrix A and using the equalities (26) and (35), we have

$$Ar^{(j)} = DBr^{(j)} = Dz^{(j)} = \overline{b_{j-1}}^{-1} De^{(j)} = e^{(j)}$$

(note that $De^{(j)} = \overline{b_{j-1}} e^{(j)}$, which follows from (3)). Further,

$$A^{-1} e^{(j)} = r^{(j)}.$$

Since $A^{-1} e^{(j)} = X^{(j)}$, then $X^{(j)} = r^{(j)}$. The components of the vector $r^{(j)}$ are given in (25). Thus,

$$x_{ij} = \bar{t}\, \overline{\mu_j} \nu_i, \;\; i = 1, 2, \ldots, j-1 \quad \text{and} \quad x_{ij} = t \mu_i \overline{\nu_j}, \;\; i = j, j+1, \ldots, n. \tag{36}$$

Combining formulas (36) with those of (12) and (18) yields

$$x_{ij} = \begin{cases} \bar{t}\, \overline{\mu_j} \nu_i, \;\; i = 1, 2, \ldots, j-1, \\ t \mu_i \overline{\nu_j}, \;\; i = j, j+1, \ldots, n \end{cases} \qquad \text{for } j = 1, 2, \ldots, n. \tag{37}$$

Note the following. Since the matrix A^{-1} is also Hermitian, then in reality we only need to calculate the lower triangular part of this matrix.

Summarizing the considerations of Sections 2 and 3, let us write the following procedure to calculate the elements of the inverse matrix $A^{-1} = [x_{ij}]_{n \times n}$ for nonsingular matrix A given in (1).

Procedure Inv 3d Hermitian

1. Input elements $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_{n-1}$ of the matrix A (see (1)).
2. Calculate the quantities f_i, g_i, p and q (see (5)):

$$f_i = \frac{a_i}{\overline{b_{i-1}}}, \; g_i = \frac{b_i}{\overline{b_{i-1}}}, \;\; i = 2, 3, \ldots, n-1; \;\; p = \frac{a_1}{b_1}, \;\; q = \frac{a_n}{\overline{b_{n-1}}}.$$

3. Calculate recursively the quantities μ_i (see (7)):

$$\mu_n = 1\,, \;\; \mu_{n-1} = -q\,,$$
$$\mu_i = -f_{i+1} \mu_{i+1} - g_{i+1} \mu_{i+2}\,, \;\; i = n-2, n-3, \ldots, 1.$$

4. Calculate recursively the quantities ν_i (see (14)):

$$\nu_1 = 1\,, \;\; \nu_2 = -p\,,$$
$$\nu_i = -\frac{1}{g_{i-1}} (\nu_{i-2} + f_{i-1} \nu_{i-1})\,, \;\; i = 3, 4, \ldots, n.$$

5. Calculate the quantity t (see (10) and Remark 1):

$$t = (a_1 \mu_1 + b_1 \mu_2)^{-1}.$$

6. Calculate the lower triangular part of the matrix A^{-1} (see (37)):

$$x_{ij} = t \mu_i \overline{\nu_j}, \; i = j, j+1, \ldots, n\,; \quad j = 1, 2, \ldots, n\,.$$

7. Set the upper triangular part of the matrix A^{-1} (see (37)):

$$x_{ij} = \overline{x_{ji}},\ i = 1, 2, \ldots, j-1\,; \quad j = 2, 3, \ldots, n\,.$$

8. Output the matrix $A^{-1} = [x_{ij}]_{n\times n}$.

End procedure

The procedure **Inv 3d Hermitian** can be useful for the following purposes. Firstly, it can be used as a basis of numerical algorithms for inverting nonsingular tridiagonal Hermitian matrices. In this case, it is easy to make sure that computing the lower triangular part of the matrix A^{-1} requires $0.5n^2 + O(n)$ arithmetical operations with complex numbers. Secondly, for matrices of special types, the procedure can be used for deriving closed form expressions for the elements of inverse matrices. The next section is devoted to this issue.

4. Toeplitz Tridiagonal Hermitian Matrices

Let us consider a matrix

$$A = \begin{bmatrix} a & b & & & \\ \bar{b} & a & b & & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & \bar{b} & a & b \\ & & & \bar{b} & a \end{bmatrix} \tag{38}$$

of order n, where a is a real number and $b \neq 0$. Additionally, we assume that

$$|a| \geq 2|b|. \tag{39}$$

Condition (39) ensures the nonsingularity of the matrix (38) (see [11], for instance).

For the matrix we are considering, the quantities calculated in Item 2 of the procedure **Inv 3d Hermitian** are as follows:

$$f_i = \frac{a}{\bar{b}},\ g_i = \frac{b}{\bar{b}},\ i = 2, 3, \ldots, n-1;\ p = \frac{a}{b},\ q = \frac{a}{\bar{b}}.$$

Further, in Item 3 of the procedure, the quantities μ_i are calculated. In our case, we have second-order recurrent relations

$$\bar{b}\mu_i + a\mu_{i+1} + b\mu_{i+2} = 0\,,\ i = n-2, n-3, \ldots, 1,$$

where $\mu_n = 1$, $\mu_{n-1} = -a/\bar{b}$. The solution of this problem is well known (see [2, 6], for instance). As a result of calculations, we get that

$$\mu_i = (-1)^{n-i}\,\frac{\bar{b}}{r}\left[\left(\frac{a+r}{2\bar{b}}\right)^{n+1-i} - \left(\frac{a-r}{2\bar{b}}\right)^{n+1-i}\right],\ i = 1, 2, \ldots, n \quad \text{if } |a| > 2|b| \tag{40}$$

and

$$\mu_i = (-1)^{n-i}\,(n+1-i)\left(\frac{a}{2b}\right)^{i-n},\ i = 1, 2, \ldots, n \quad \text{if } |a| = 2|b|, \tag{41}$$

where

$$r \equiv \sqrt{a^2 - 4|b|^2}.$$

In a similar way, we find expressions for the quantities ν_i determined in Item 4 of the procedure. These quantities satisfy the following second-order recurrent relations:

$$\bar{b}\nu_{i-2} + a\nu_{i-1} + b\nu_i = 0\,, \; i = 3, 4, \dots, n,$$

where $\nu_1 = 1$, $\nu_2 = -a/b$. Making calculations, we find that

$$\nu_i = (-1)^{i-1}\,\frac{b}{r}\left[\left(\frac{a+r}{2b}\right)^i - \left(\frac{a-r}{2b}\right)^i\right], \; i = 1, 2, \dots, n \quad \text{if } |a| > 2|b| \tag{42}$$

and

$$\nu_i = (-1)^{i-1}\, i \left(\frac{a}{2b}\right)^{i-1}, \; i = 1, 2, \dots, n \quad \text{if } |a| = 2|b|. \tag{43}$$

In Item 5 of the procedure, the quantity t is calculated. Using the expressions (40) and (41), we get

$$t = (-1)^{n-1}\frac{r}{\bar{b}^2}\left[\left(\frac{a+r}{2\bar{b}}\right)^{n+1} - \left(\frac{a-r}{2\bar{b}}\right)^{n+1}\right]^{-1} \quad \text{if } |a| > 2|b| \tag{44}$$

and

$$t = \frac{(-1)^{n-1}}{n+1}\,\frac{2}{a}\left(\frac{a}{2b}\right)^{n-1} \quad \text{if } |a| = 2|b|. \tag{45}$$

Finally, in Items 6 and 7 of the procedure, the elements x_{ij} of the inverse matrix A^{-1} are calculated. If $|a| > 2|b|$, then we use the formulas (40), (42) and (44). For the values $j = 1, 2, \dots, n$, we obtain that

$$x_{ij} = \frac{(-1)^{j-i}}{r}\,\frac{\left[\left(\frac{a+r}{2b}\right)^i - \left(\frac{a-r}{2b}\right)^i\right]\left[\left(\frac{a+r}{2b}\right)^{n+1-j} - \left(\frac{a-r}{2b}\right)^{n+1-j}\right]}{\left[\left(\frac{a+r}{2b}\right)^{n+1} - \left(\frac{a-r}{2b}\right)^{n+1}\right]} \tag{46}$$

if $i = 1, 2, \dots, j-1$ and

$$x_{ij} = \frac{(-1)^{i-j}}{r}\,\frac{\left[\left(\frac{a+r}{2\bar{b}}\right)^{n+1-i} - \left(\frac{a-r}{2\bar{b}}\right)^{n+1-i}\right]\left[\left(\frac{a+r}{2\bar{b}}\right)^j - \left(\frac{a-r}{2\bar{b}}\right)^j\right]}{\left[\left(\frac{a+r}{2\bar{b}}\right)^{n+1} - \left(\frac{a-r}{2\bar{b}}\right)^{n+1}\right]} \tag{47}$$

if $i = j, j+1, \dots, n$. As an example, consider the matrix

$$A = \begin{bmatrix} 5 & 2\mathtt{i} & & & \\ -2\mathtt{i} & 5 & 2\mathtt{i} & & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & -2\mathtt{i} & 5 & 2\mathtt{i} \\ & & & -2\mathtt{i} & 5 \end{bmatrix}.$$

According to the expressions (46) and (47) we find that

$$x_{ij} = \begin{cases} \dfrac{(2^i - 2^{-i})(2^{n+1-j} - 2^{-n-1+j})}{3(2^{n+1} - 2^{-n-1})}\,\mathtt{i}^{i-j}, & i = 1, 2, \dots, j-1, \\[2ex] \dfrac{(2^{n+1-i} - 2^{-n-1+i})(2^j - 2^{-j})}{3(2^{n+1} - 2^{-n-1})}\,\mathtt{i}^{i-j}, & i = j, j+1, \dots, n, \end{cases}$$

where the symbol $\mathtt{i}$ stands for the imaginary unit.

Now consider the case $|a| = 2|b|$. For the values $j = 1, 2, \ldots, n$, using the formulas (41), (43) and (45), we find that

$$x_{ij} = \begin{cases} (-1)^{j-i} \dfrac{(n+1-j)i}{n+1} \dfrac{2}{a} \left(\dfrac{a}{2b}\right)^{i-1} \left(\dfrac{a}{2\bar{b}}\right)^{j-1}, & i = 1, 2, \ldots, j-1, \\ (-1)^{i-j} \dfrac{(n+1-i)j}{n+1} \dfrac{2}{a} \left(\dfrac{a}{2b}\right)^{i-1} \left(\dfrac{a}{2\bar{b}}\right)^{j-1}, & i = j, j+1, \ldots, n. \end{cases} \tag{48}$$

For the matrix

$$A = \begin{bmatrix} 2 & \mathtt{i} & & & \\ -\mathtt{i} & 2 & \mathtt{i} & & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & -\mathtt{i} & 2 & \mathtt{i} \\ & & & -\mathtt{i} & 2 \end{bmatrix},$$

the expressions (48) take the following form:

$$x_{ij} = \begin{cases} (-1)^j \dfrac{(n-j+1)i}{n+1} \mathtt{i}^{i+j}, & i = 1, 2, \ldots, j-1, \\ (-1)^j \dfrac{(n-i+1)j}{n+1} \mathtt{i}^{i+j}, & i = j, j+1, \ldots, n, \end{cases} \qquad j = 1, 2, \ldots, n.$$

5. Conclusion

In this paper, we have constructed the computational procedure **Inv 3d Hermitian** for inversion of tridiagonal Hermitian matrices. This procedure can be used as a numerical algorithm with an optimal number of arithmetic operations (see the comment on the procedure at the end of Section 3). In certain cases, the procedure can also be used to derive closed-form expressions for the elements of inverse matrices. In this regard, Toeplitz tridiagonal Hermitian matrices in Section 4 were considered.

References

[1] G. H. Golub and Ch. F. van Loan, *Matrix Computations*, The John Hopkins University Press, 1996.

[2] D. Kincaid and W. Cheney, *Numerical Analysis*, Brooks/Cole, Pacific Grove, CA, 1991.

[3] D. S. Watkins, *Fundamentals of matrix computations*, A Wiley Intercience Publ., 2010.

[4] B. Buchberger and G. A. Yemel'yanenko, "Methods for inverting tridiagonal matrices", *J. Comput. Math. and Math. Physics*, vol. 13, No.3, pp. 546-554, 1973 (in Russian).

[5] M. El-Mikkawy and A. Karawia, "Inversion of general tridiagonal matrices", *Applied Math. Letters*, vol. 19, pp. 712-720, 2006.

[6] V.P. Il'in and Yu. I. Kuznetsov, *Tridiagonal Matrices and Their Applications*, (in Russian), Nauka, 1985.

[7] G.Y. Hu and R.F. O'Connell, "Analytical inversion of symmetric tridiagonal matrices", *J. Phys. A: Math. Gen.*, vol. 29, pp. 1511-1513, 1996.

[8] Y. Huang and W.F. McColl, "Analytic inversion of general tridiagonal matrices", *J. Phys. A: Math. Gen.*, vol. 30, pp. 7919-7933, 1997.

[9] J. W. Lewis, "Inversion of tridiagonal matrices", *Numer. Math.*, vol. 38, pp. 333-345, 1982.

[10] R. A. Usmani, "Inversion of Jacobi's tridiagonal matrix", *Computers Math. Applic.*, vol. 27, no. 8, pp. 59-66, 1994.

[11] R. Horn and Ch. Johnson, *Matrix Analysis*, Cambridge University Press, 1986.

Երեքանկյունագծային հերմիտյան մատրիցների անալիտիկ հակադարձում

Յուրի Ռ. Հակոբյան և Ավետիք Հ. Մանուկյան

Երևանի պետական համալսարան, Երևան, Հայաստան
e-mail: yuri.hakopian@ysu.am, avetiq.manukyan1@ysumail.am

Ամփոփում

Հոդվածում տրվում է երեքանկյունագծային հերմիտյան մատրիցների հակադարձման ալգորիթմը, որի թվային իրականացումը պահանջում է օպտիմալ թվով թվաբանական գործողություններ: Հաշվողական պրոցեդուրան իրենից ներկայացնում է հակադարձ մատրիցի տարրերի հաշվմանը հանգեցնող անդրադարձ առնչությունների հաջորդականության: Հատուկ տիպի մատրիցների համար և, մասնավորապես, տյոպլիցյան երեքանկյունագծային հերմիտյան մատրիցների համար, ստացված առնչությունները հանգեցնում են հակադարձ մատրիցի տարրերի համար բացահայտ բանաձևերի:

Բանալի բառեր՝ հակադարձ մատրից, երեքանկյունագծային մատրից, հերմիտյան մատրից, տյոպլիցյան մատրից:

Аналитическое обращение трехдиагональных изображени

Юрий Р. Акопян и Аветик А. Манукян

Ереванский государственный университет, Ереван, Армения
e-mail: yuri.hakopian@ysu.am, avetiq.manukyan1@ysumail.am

Аннотация

В статье дается алгоритм обращения трехдиагональных эрмитовых матриц, численная реализация которого осуществляется за оптимальное число арифметических операций. Вычислительная процедура представляет собой последовательность рекуррентных соотношений, приводящих к вычислению элементов обратной матрицы. Для матриц специального типа и, в частности, для тёплицевых трехдиагональных эрмитовых матриц, полученные соотношения приводят к явным формулам для элементов обратной матрицы.

Ключевые слова: обратная матрица, трехдиагональная матрица, эрмитова матрица, тёплицева матрица.

Mathematical Problems of Computer Science 58, 20–31, 2022.
doi:10.51408/1963-0089

UDC 519.1

On an Extension of the Ghouila-Houri Theorem

Samvel Kh. Darbinyan

Institute for Informatics and Automation Problems of NAS RA
e-mail: samdarbin@iiap.sci.am

Abstract

Let D be a 2-strong digraph of order $n \geq 8$ such that for every vertex $x \in \mathcal{V}(\mathcal{D})\backslash\{z\}$, $d(x) \geq n$ and $d(z) \geq n-4$, where z is a vertex in $\mathcal{V}(\mathcal{D})$. We prove that:

If D contains a cycle passing through z of length equal to $n-2$, then D is Hamiltonian.

We also give a new sufficient condition for a digraph to be Hamiltonian-connected.
Keywords: Digraphs, Hamiltonian cycles, Hamiltonian-connected, 2-strong.
Article info: Received 21 April 2022; received in revised form 16 September 2022; accepted 15 November 2022.
Acknowledgement: We thank the referees for their valuable comments and suggestions that improved the presentation considerably.

1. Introduction

In this paper, we consider finite digraphs (directed graphs) without loops and multiple arcs. The order of a digraph D is the number of its vertices. We shall assume that the reader is familiar with the standard terminology on digraphs. Terminology and notations not described below follow [1]. Every cycle and path is assumed simple and directed. A cycle (path) in a digraph D is called *Hamiltonian* (*Hamiltonian path*) if it includes every vertex of D. A digraph D is *Hamiltonian* if it contains a Hamiltonian cycle, and it is *Hamiltonian-connected* if for any pair of ordered vertices x and y there exists a Hamiltonian path from x to y.

There are numerous sufficient conditions for the existence of a Hamiltonian cycle in a digraph (see, [1]–[3]). Let us recall the following sufficient conditions for a digraph to be Hamiltonian.

Theorem 1: (Ghouila-Houri [4]). *Let D be a strong digraph of order $n \geq 2$. If for every vertex $x \in \mathcal{V}(\mathcal{D})$, $d(x) \geq n$, then D is Hamiltonian.*

Theorem 2: (Meyniel [5]). *Let D be a strong digraph of order $n \geq 2$. If $d(x)+d(y) \geq 2n-1$ for all pairs of non-adjacent vertices x and y in D, then D is Hamiltonian.*

Nash-Williams [6] raised the problem of describing all the extreme digraphs in Theorem 1, that is, all digraphs with minimum degree at least $|D|-1$, that do not have a Hamiltonian cycle. As a solution to this problem, Thomassen [7] proved a structural theorem on the extreme digraphs. An analogous problem for Theorem 2 was considered by the author [8]. In [8], we generalize Thomassen's structural theorem (Theorem 1, in [7]), characterizing the non-Hamiltonian strong digraphs of order n with the degree condition that $d(x) + d(y) \geq 2n - 2$ for every pair of non-adjacent distinct vertices x, y. Moreover, in [8], it was also proved that if m is the length of a longest cycle in D, then D contains cycles of all lengths $k = 2, 3, \ldots, m$. The following conjecture was suggested by Thomassen.

Conjecture 1: (Thomassen [9], see Conjecure 1.6.7 in [2]). *Every 3-strong digraph of order n and with minimum degree at least $n + 1$ is Hamiltonian-connected.*

In [10], we disprove this conjecture, by proving the following three theorems.

Theorem 3: *Every k-strong ($k \geq 1$) digraph of order n, which has $n - 1$ vertices of degrees at least n, is Hamiltonian if and only if any $(k + 1)$-strong digraph of order $n + 1$ with minimum degree at least $n + 2$ is Hamiltonian-connected.*

Theorem 4: *For every $n \geq 8$, there is a non-Hamiltonian 2-strong digraph D of order n with minimum degree equal to 4 such that D has $n - 1$ vertices of degrees at least n.*

Theorem 5: *For every $n \geq 9$, there exists a 3-strong digraph D of order n with minimum degree at least $n+1$ such that D contains two distinct vertices u, v for which $u \leftrightarrow v$, $d^+_D(u) + d^-_D(v) = 6$ and there is no (u, v)-Hamiltonian path.*

In view of Theorems 4, 5 and Conjecture 1, it is natural to pose the following problem.

Problem: *Let D be a 2-strong digraph of order $n \geq 9$. Suppose that $n - 1$ vertices of D have degrees at least n and a vertex x has degree is at least $n - m$, where $1 \leq m \leq n - 5$. Find the maximum value of m, for which D is Hamiltonian.*

Goldberg, Levitskaya and Satanovskiy [11] relaxed the conditions of the Ghouila-Houri theorem. They proved the following theorem.

Theorem 6: (Goldberg et al. [11]). *Let D be a strong digraph of order $n \geq 2$. If for every vertex $x \in \mathcal{V}(D) \setminus \{z\}$, $d(x) \geq n$ and $d(z) \geq n - 1$, then D is Hamiltonian.*

Note that Theorem 6 is an immediate consequence of Theorem 2. In [11], the authors for any $n \geq 5$ presented two examples of non-Hamiltonian strong digraphs of order n such that:

(i) In the first example, $n - 2$ vertices have degrees equal to $n + 1$ and the other two vertices have degrees equal to $n - 1$.

(ii) In the second example, $n-1$ vertices have degrees at least n and the remaining vertex has degree equal to $n - 2$.

In [12], it was reported that the following theorem holds.

Theorem 7: (Darbinyan [12]). *Let D be a 2-strong digraph of order $n \geq 9$ with minimum degree at least $n - 4$. If $n - 1$ vertices of D have degrees at least n, then D is Hamiltonian.*

In this article, we present the first part of the proof of Theorem 7, which we formulate as Theorem 9. The proof of the last theorem has never been published. It is worth mentioning that the proof presented here differs from the previous handwritten proof and is significantly shorter and more general than the previous one. The second part of the proof (i.e., the complete proof) of Theorem 7 we will present in the forthcoming paper, where we also present two examples of digraphs, which show that the bounds $n \geq 9$ and $n-4$ in Theorem 7 are sharp in a sense.

2. Further Terminology and Notation

For the sake of clarity we repeat the most impotent definition. The vertex set and the arc set of a digraph D are denoted by $\mathcal{V(D)}$ and $\mathcal{A(D)}$, respectively. The order of a digraph D is the number of its vertices. The converse digraph of D is the digraph obtained from D by reversing the direction of all arcs. The arc of a digraph D directed from x to y is denoted by xy or $x \to y$ (we also say that x *dominates* y or y is an *out-neighbour* of x and x is an *in-neighbour* of y), and $x \leftrightarrow y$ denotes that $x \to y$ and $y \to x$ ($x \leftrightarrow y$ is called *2-cycle*). If $x \to y$ and $y \to z$, we write $x \to y \to z$. If A and B are two disjoint subsets of $\mathcal{V(D)}$ such that every vertex of A dominates every vertex of B, then we say that A *dominates* B, denoted by $A \to B$. We define $\mathcal{A}(A \to B) = \{xy \in \mathcal{A}(D) \,|\, x \in A, y \in B\}$ and $\mathcal{A}(\mathcal{A}, \mathcal{B}) = \mathcal{A}(\mathcal{A} \to \mathcal{B}) \cup \mathcal{A}(\mathcal{B} \to \mathcal{A})$. If $x \in \mathcal{V(D)}$ and $A = \{x\}$ we sometimes write x instead of $\{x\}$. Let $N_D^+(x)$, $N_D^-(x)$ denote the set of out-neighbors, respectively the set of in-neighbors of a vertex x in a digraph D. If $A \subseteq \mathcal{V(D)}$, then $N_D^+(x, A) = A \cap N_D^+(x)$ and $N_D^-(x, A) = A \cap N_D^-(x)$. The *out-degree* of x is $d_D^+(x) = |N_D^+(x)|$ and $d_D^-(x) = |N_D^-(x)|$ is the *in-degree* of x. Similarly, $d_D^+(x, A) = |N_D^+(x, A)|$ and $d_D^-(x, A) = |N_D^-(x, A)|$. The *degree* of the vertex x in D is defined as $d_D(x) = d_D^+(x) + d_D^-(x)$ (similarly, $d_D(x, A) = d_D^+(x, A) + d_D^-(x, A)$). We omit the subscript if the digraph is clear from the context. The subdigraph of D induced by a subset A of $\mathcal{V(D)}$ is denoted by D. In particular, $D - A = D\langle \mathcal{V(D)} \setminus \mathcal{A} \rangle$. For integers a and b, $a \leq b$, by $[a, b]$ we denote the set $\{x_a, x_{a+1}, \ldots, x_b\}$. If $j < i$, then $\{x_i, \ldots, x_j\} = \emptyset$.

The path (respectively, the cycle) consisting of the distinct vertices $x_1, x_2, \ldots, x_m$ ($m \geq 2$) and the arcs $x_i x_{i+1}$, $i \in [1, m-1]$ (respectively, $x_i x_{i+1}$, $i \in [1, m-1]$, and $x_m x_1$), is denoted by $x_1 x_2 \cdots x_m$ (respectively, $x_1 x_2 \cdots x_m x_1$). The *length* of a cycle or a path is the number of its arcs. Let D be a digraph and $z \in \mathcal{V(D)}$. By $C_m(z)$ (respectively, $C(z)$) we denote a cycle in D of length m (respectively, any cycle in D), which contains the vertex z. We say that $P = x_1 x_2 \cdots x_m$ is a path from x_1 to x_m or is an (x_1, x_m)*-path*. A digraph D is *strong* (*strongly connected*) if, for every pair x, y of distinct vertices in D, there exists an (x, y)-path and a (y, x)-path. A digraph D is k*-strong* (k*-strongly connected*) if, $|\mathcal{V(D)}| \geq \| + \infty$ and for any set A of at most $k-1$ vertices $D - A$ is strong. Two distinct vertices x and y are *adjacent* if $xy \in$ or $yx \in \mathcal{A(D)}$ (or both). The *converse* digraph of D is the digraph obtained from D by replacing the direction of all arcs. We will use *the principle of digraph duality*: Let D be a digraph, then D contains a subdigraps H if and only if the converse digraph of D contain the converse of subdigraph H.

3. Preliminaries

In our proofs, we will use the following well-known simple lemma.

Lemma 1: (Häggkvist and Thomassen [13]). *Let D be a digraph of order $n \geq 3$ containing a cycle C_m of length m, $m \in [2, n-1]$. Let x be a vertex not contained in this cycle. If $d(x, \mathcal{V}(C_m)) \geq m+1$, then for every $k \in [2, m+1]$, D contains a cycle C_k including x.*

The next lemma is a slight modification of a lemma by Bondy and Thomassen [14], it is very useful and will be used extensively throughout this paper.

Lemma 2:. *Let D be a digraph of order $n \geq 3$ containing a path $P := x_1x_2 \dots x_m$, $m \in [2, n-1]$. Let x be a vertex not contained in this path. If one of the following condition holds:*

(i) $d(x, \mathcal{V}(P)) \geq m + 2$,

(ii) $d(x, \mathcal{V}(P)) \geq m + 1$ and $xx_1 \notin \mathcal{A}(D)$ or $x_mx \notin \mathcal{A}(\mathcal{D})$,

(iii) $d(x, \mathcal{V}(P)) \geq m$ and $xx_1 \notin \mathcal{A}(\mathcal{D})$ and $x_mx \notin \mathcal{A}(\mathcal{D})$,

then there is an $i \in [1, m-1]$ such that $x_i \to x \to x_{i+1}$, i.e., D contains a path $x_1x_2 \dots x_ixx_{i+1} \dots x_m$ of length m (we say that x can be inserted into P).

Using Lemma 2, we can prove the following lemma.

Lemma 3: *Let $P := x_1x_2 \dots x_m$, $m \in [3, n-1]$, be a longest (x_1, x_m)-path in a digraph D of order n. Suppose that $y \in \mathcal{V}(D) \setminus \mathcal{V}(P)$ and there is no $i \in [1, m-2]$ such that $x_i \to y \to x_{i+2}$. Then the following holds:*

(i) If $yx_1 \notin \mathcal{A}(\mathcal{D})$, $x_1y \in \mathcal{A}(\mathcal{D})$ and $d(y, \mathcal{V}(P)) \geq m$, then $d(y, \mathcal{V}(P)) = m$ and $\{x_1, x_2, \dots, x_m\} \to y$;

(ii) If $x_my \notin \mathcal{A}(\mathcal{D})$, $yx_m \in \mathcal{A}(\mathcal{D})$ and $d(y, \mathcal{V}(P)) \geq m$, then $d(y, \mathcal{V}(P)) = m$ and $y \to \{x_1, x_2, \dots, x_m\}$;

(iii) If $d(y, \mathcal{V}(P)) \geq m+1$, then $d(y, \mathcal{V}(P)) = m+1$ and there exists an integer $q \in [1, m]$ such that $\{x_q, x_{q+1}, \dots, x_m\} \to y \to \{x_1, x_2, \dots, x_q\}$.

Proof. To prove the lemma, it suffices to show that every vertex $x_i \in \mathcal{V}(\mathcal{P})$ is adjacent to y. Assume that this is not the case. (i) Let y and x_t be not adjacent. Then $t \geq 2$ since $x_1 \to y$. Since P is a longest (x_1, x_m)-path, we have that y cannot be inserted into P. Using Lemma 2(ii) and the assumption that $yx_1 \notin \mathcal{A}(\mathcal{D})$, we obtain $x_my \in \mathcal{A}(\mathcal{D})$, $2 \leq t \leq m-1$ and

$$m \leq d(y, \mathcal{V}(P)) = d(y, \{x_1, \dots, x_{t-1}\}) + d(y, \{x_{t+1}, \dots, x_m\}) \leq t - 1 + (m - t + 1) = m.$$

This means that $d(y, \{x_1, \dots, x_{t-1}\}) = t - 1$ and $d(y, \{x_{t+1}, \dots, x_m\}) = m - t + 1$. Again using Lemma 2, we obtain that $x_{t-1} \to y \to x_{t+1}$, which contradicts the supposition of Lemma 3. This contradiction shows that every vertex x_i is adjacent to y.

In a similar way, one can show that if (ii) or (iii) holds, then every vertex of P also is adjacent to y. Lemma 3 is proved. □

In [10], the author proved the following theorem.

Theorem 8: (Darbinyan [12]). *Let D be a strong digraph of order $n \geq 3$. Suppouse that $d(x) + d(y) \geq 2n - 1$ for all pairs of non-adjacent vertices $x, y \in \mathcal{V}(D) \setminus \{z\}$, where z is some vertex in $\mathcal{V}(\mathcal{D})$. Then D is Hamiltonian or contains a cycle of length $n-1$.*

Using Theorem 8 and Lemmas 1 and 2, it is not difficult to show that the following corollaries are true.

Corollary 1: *Let D be a strong digraph of order $n \geq 3$ satisfying the condition of Theorem 8. Then D has a cycle that contains all the vertices of D maybe except z.*

Corollary 2: *Let D be a strong digraph of order $n \geq 3$. Suppose that $n-1$ vertices of D have degrees at least n. Then D is Hamiltonian or contains a cycle of length $n-1$ (in fact, D has a cycle that contains all the vertices of degrees at least n).*

In this section, we also will prove the following lemma. We will use this lemma in the second part of the proof of Theorem 7.

Lemma 4: *Let D be a digraph of order $n \geq 4$ such that for any vertex $x \in \mathcal{V}(D)\setminus\{z\}$, $d(x) \geq n$ and $d(z) \leq n-2$, where z is some vertex in $\mathcal{V}(\mathcal{D})$. Suppose that $C_m(z) = x_1x_2\ldots x_mx_1$ with $m \leq n-2$ is a longest cycle through z. If $D\langle V(D)\setminus V(C_m(z))\rangle$ is strong and D contains a $C_m(z)$-bypass $P = x_iy_1y_2\ldots y_lx_j$ such that $|\mathcal{V}(C_m(z)[x_{i+1}, x_{j-1}])|$ is smallest possible over all $C_m(z)$-bypasses, then $z \in \mathcal{V}(C_m(z)[x_{i+1}, x_{j-1}])$.*

Proof. Without loss of generality, we assume that $x_j = x_1$, $x_i = x_{m-k}$, $k \geq 1$, $\mathcal{A}(\{y_1,\ldots,y_l\}, \mathcal{V}(C_m(z)[x_{m-k+1}, x_m])) = \emptyset$ and k is minimum possible with this property over all $C_m(z)$-bypasses. Extending the path $C_m(z)[x_1, x_{m-k}]$ with the vertices of $\mathcal{V}(C_m(z)[x_{m-k+1}, x_m])$ as much as possible, we obtain an (x_1, x_{m-k})-path, say R. Since $C_m(z)$ is a longest cycle through z, some vertices $u_1, u_2, \ldots, u_d \in \mathcal{V}(C_m(z)[x_{m-k+1}, x_m])$, $1 \leq d \leq k$, are not on the obtained extended path R. Using Lemma 2, we obtain that $d(y_i, \mathcal{V}V(C_m(z))) \leq m-k+1$ and $d(u_i, \mathcal{V}(C_m(z))) \leq m+d-1$. Put $B := \mathcal{V}(D) \setminus (\mathcal{V}(C_m(z)) \cup \mathcal{V}(\mathcal{P}))$. Note that $|B| = n-m-l$. Let v be an arbitrary vertex in B. From the minimality of k, we have that D contains no paths of the types $u_i \to v \to y_j$ and $y_j \to v \to u_i$, which in turn implies that $d^+(u_i, B) + d^-(y_j, B) \leq |B|$ and $d^-(u_i, B) + d^+(y_j, B) \leq |B|$. Therefore, $d(u_i, B) + d(y_j, B) \leq 2|B| = 2(n-m-l)$. Thus, we have

$$\begin{aligned} d(u_i) + d(y_j) &= d(u_i, \mathcal{V}(C_m(z))) + d(y_j, \mathcal{V}(C_m(z))) + d(u_i, B) + d(y_j, B) + d(y_j, \{y_1, \ldots, y_l\}) \\ &\leq m+d-1+m-k+1+2n-2m-2l+2l-2 = 2n-2-(k-d) \leq 2n-2. \end{aligned}$$

This is possible if $u_i = z$. Therefore, $d = 1$ and $z \in \mathcal{V}(C_m(z)[x_{m-k+1}, x_m])$. Lemma 4 is proved. □

4. The Main Result

In this section, we prove the following theorem.

Theorem 9: *Let D be a 2-strong digraph of order $n \geq 8$. Suppose that for every $x \in \mathcal{V}(D) \setminus \{z\}$, $d(x) \geq n$ and $d(z) \geq n-4$, where z is a vertex in $\mathcal{V}(\mathcal{D})$. If D contains a cycle of length $n-2$ passing through z (i.e., a cycle $C_{n-2}(z)$), then D is Hamiltonian.*

Before we prove our main result, we will prove the following lemma.

Lemma 5: *Let D be a non-Hamiltonian 2-strong digraph of order n such that for any vertex $x \in \mathcal{V}(D) \setminus \{z\}$, $d(x) \geq n$ and $d(z) \leq n-2$, where z is an arbitrary fixed vertex in $\mathcal{V}(\mathcal{D})$. Suppose that $C_{m+1}(z) = x_1x_2\ldots x_mzx_1$ with $m \in [2, n-3]$ is a longest cycle in D, $d(z, Y) = 0$ and $D\langle Y\rangle$ is a strong digraph, where $Y := \mathcal{V}(D) \setminus \mathcal{V}(C_{m+1}(z))$. Let y_1, y_2 be two distinct vertices in Y. If for each $y_i \in \{y_1, y_2\}$, $d(y_i, \{x_1, x_2, \ldots, x_m\}) = m+1$, then $n \geq 6$ and $d(z) \leq m-2$.*

Proof. By contradiction, suppose that $d(z) \geq m-1$. We denote by P the path $x_1x_2\ldots x_m$. Note that $|Y| = n-m-1$. Since the path P cannot be extended with any vertex $y \in Y$, by Lemma 2, $d(y, \mathcal{V}(P)) \leq m+1$ and

$$n \leq d(y) = d(y, \mathcal{V}(P)) + d(y, Y) \leq m+1+d(y, Y),\ d(y, Y) \geq n-m-1 = |Y|. \tag{1}$$

Since D is 2-strong and $C_{m+1}(z)$ is a longest cycle, using Lemma 2 and $d(y_i, \mathcal{V}(P)) = m+1$ it is not difficult to show that there is an integer $l \in [2, m-1]$ such that

$$\{x_l, x_{l+1}, \ldots, x_m\} \to \{y_1, y_2\} \to \{x_1, x_2, \ldots, x_l\}. \tag{2}$$

Since $d(y,Y) \geq n-m-1 = |Y|$(by (1)), and $D\langle Y\rangle$ is strong, by the Ghouila-Houri theorem, $D\langle Y\rangle$ is Hamiltonian. Put $E := \{x_1, x_2, \dots, x_{l-1}\}$ and $F := \{x_{l+1}, x_{l+2}, \dots, x_m\}$. Since $C_{m+1}(z)$ is a longest cycle and $D\langle Y\rangle$ is strong, from (2) it follows that

$$\mathcal{A}(\mathcal{E} \to \mathcal{Y}) = \mathcal{A}(\mathcal{Y} \to \mathcal{F}) = \emptyset. \tag{3}$$

Note that from $|Y| \geq 2$, $|E| \geq 1$ and $|F| \geq 1$ it follows that $n \geq 6$. We need to prove the following Claims 1-2 bellow.

Claim 1.
(i) If $d^-(z,E) \geq 1$, then $d^+(z,F) = 0$.
(ii) $\mathcal{A}(\mathcal{E} \to \mathcal{F}) \neq \emptyset$.

Proof. (i) By contradiction, suppose that $x_i \in E$, $x_j \in F$ and $x_i \to z \to x_j$. Then by (2), $y_1 \to x_{i+1}$ and $x_{j-1} \to y_2$. Hence, $C_{m+3}(z) = x_1x_2 \dots x_izx_j \dots x_my_1x_{i+1} \dots x_{j-1}y_2x_1$, a contradiction.

(ii) Suppose, on the contrary, that $\mathcal{A}(\mathcal{E} \to \mathcal{F}) = \emptyset$. Then using Claim 1(i) and (3), we obtain: if $d^-(z,E) \geq 1$, then $d^+(z,F) = 0$ and $\mathcal{A}(E \cup Y \cup \{z\} \to F) = \emptyset$, if $d^-(z,E) = 0$, then $\mathcal{A}(E \cup Y \to F \cup \{z\}) = \emptyset$. Therefore, $D - x_l$ is not strong, which contradicts that D is 2-strong. □

From now on, we assume that $x_ax_b \in \mathcal{A}(\mathcal{E} \to \mathcal{F})$. Note that $1 \leq a \leq l-1$ and $l+1 \leq b \leq m$. We may assume that b is the maximum and a is the minimum with these properties. By (2), we have

$$x_{b-1} \to \{y_1, y_2\} \to x_{a+1}. \tag{4}$$

Since z cannot be inserted into P, using Lemma 2(ii) and Clam 1(i), we obtain

$$d(z, \{x_1, x_2, \dots, x_a\}) + d(z, \{x_b, x_{b+1}, \dots, x_m\}) \leq a + m - b + 2. \tag{5}$$

By $R(y_i, y_{3-i})$, where $i \in [1,2]$, we denote a longest (y_i, y_{3-i})-path in $D\langle Y\rangle$. From now on, assume that $R(y_i, y_{3-i}) = R(y_1, y_2)$.

Claim 2.
(i) *If $i \in [a+1, l-1]$, then $x_iz \notin \mathcal{A}(\mathcal{D})$.*
(ii) If $j \in [l+1, b-1]$, then $zx_j \notin \mathcal{A}(\mathcal{D})$.
(iii) If $i \in [a+1, l]$ and $i - a \leq 2$, then $zx_i \notin \mathcal{A}(\mathcal{D})$.
(iv) If $j \in [l, b-1]$ and $b - j \leq 2$, then $x_jz \notin \mathcal{A}(\mathcal{D})$.

Proof. Each of claims (i)-(iv) we prove by contradiction.

(i) Assume that $i \in [a+1, l-1]$ and $x_iz \in \mathcal{A}(\mathcal{D})$. Then by (2) and (4), we have $C_{m+3}(z) = x_1x_2 \dots x_ax_b \dots x_my_1x_{i+1} \dots x_{b-1}y_2x_{a+1} \dots x_izx_1$, a contradiction.

(ii) Assume that $j \in [l+1, b-1]$ and $zx_j \in \mathcal{A}(\mathcal{D})$. Then by (2) and (4), we have $C_{m+3}(z) = x_1x_2 \dots x_ax_b \dots x_mzx_j \dots x_{b-1}y_1x_{a+1} \dots x_{j-1}y_2x_1$, a contradiction.

(iii) Assume that $i \in [a+1, l]$, $i - a \leq 2$ and $zx_i \in \mathcal{A}(\mathcal{D})$. Then $C(z) = x_1x_2 \dots x_ax_b \dots x_mzx_i \dots x_{b-1}R(y_1, y_2)x_1$ is a cycle of length at least $m+2$, a contradiction.

(iv) Assume that $j \in [l, b-1]$, $b - j \leq 2$ and $x_jz \in \mathcal{A}(\mathcal{D})$. Then $C(z) = x_1x_2 \dots x_ax_b \dots x_mR(y_1, y_2)x_{a+1} \dots x_jzx_1$ is a cycle of length at least $m+2$, a contradiction. Claim 2 is proved. □

Now we will consider the following cases depending on the values of a and b with respect to l.

Case 1. $a \leq l-3$ and $b \geq l+3$.

Then by Claim 2, $d(z, \{x_{a+1}, x_{a+2}, x_{b-2}, x_{b-1}\}) = 0$. Therefore, since z cannot be inserted into P, using (5) and Lemma 2, we obtain

$$m-1 \leq d(z, \{x_1, x_2, \ldots, x_a, x_b, x_{b+1}, \ldots, x_m\}) + d(z, \{x_{a+3}, \ldots, x_{b-3}\})$$

$$\leq a+m-b+2+b-3-a-2+1 = m-2,$$

which is a contradiction.

Case 2. $a \leq l-3$ and $b = l+2$.

Then by Claim 2, $d(z, \{x_{a+1}, x_{a+2}, x_{l+1}\}) = 0$ and $x_l z \notin \mathcal{A}(\mathcal{D})$. Therefore, since z cannot be inserted into P, using (5) and Lemma 2, we obtain

$$m-1 \leq d(z, \{x_1, x_2, \ldots, x_a, x_b, x_{b+1}, \ldots, x_m\}) + d(z, \{x_{a+3}, \ldots, x_l\})$$

$$\leq a+m-b+2+l-a-2 = m-(l+2)+l = m-2,$$

which is a contradiction.

Case 3. $a \leq l-3$ and $b = l+1$.

Then by Claim 2, $d(z, \{x_{a+1}, x_{a+2}\}) = 0$ and $x_l z \notin \mathcal{A}(\mathcal{D})$. Similar to Case 2, we obtain

$$m-1 \leq d(z, \{x_1, x_2, \ldots, x_a, x_b, x_{b+1}, \ldots, x_m\}) + d(z, \{x_{a+3}, \ldots, x_l\})$$

$$\leq a+m-b+2+l-a-2 = m-b+l = m-(l+1) = m-1.$$

This implies that $d(z, \{x_{a+3}, \ldots, x_l\}) = l-a-2$. Hence, by Claim 2(i) and $x_l z \notin \mathcal{A}(\mathcal{D})$, $z \to \{x_{a+3}, \ldots, x_l\}$. From this and (4), we see that the cycle $Q(z) = x_1 x_2 \ldots x_a x_b \ldots x_m z$ $x_{a+3} \ldots x_l R(y_1, y_2) x_1$ has length equal to $m-1+|\mathcal{V}(R(y_1, y_2))|$. Since $C_{m+1}(z)$ is a longest cycle and $D\langle Y \rangle$ is Hamiltonian, it follows that $|\mathcal{V}(R(y_1, y_2))| = |Y| = 2$. Then $m = n-3$, $y_1 \leftrightarrow y_2$, $x_{a+1} \leftrightarrow x_{a+2}$ and x_{a+1} (x_{a+2}) is adjacent to each vertex $x_i \in \{x_1, x_2, \ldots x_m\}$, as $d(x_{a+1}) \geq n$ $(d(x_{a+2}) \geq n)$ and x_{a+1} (x_{a+2}) cannot be inserted into $Q(z)$.

We will distinguish two subcases.

Subcase 3.1. $m \geq l+2$. From the minimality of a and the maximality of b, it follows that

$$\mathcal{A}(\{x_1, x_2, \ldots, x_a\} \to \{x_{b+1}, x_{b+2}, \ldots, x_m\}) = \emptyset. \tag{6}$$

Assume that $x_i \to x_j$ with $i \in [a+1, l]$ and $j \in [l+2, m]$. Using (4) and the fact that $z x_{a+3} \in \mathcal{A}(\mathcal{D})$, it is not difficult to see that if $i \in [a+1, a+2]$, then $C(z) = x_1 x_2 \ldots x_{a+1}(x_{a+2}) x_j \ldots x_m z x_{a+3} \ldots x_{j-1} y_1 y_2 x_1$ is a cycle of length at least $m+2$, if $i \in [a+3, l-1]$, then $C_{m+3}(z) = x_1 x_2 \ldots x_i x_j \ldots x_m z x_{i+1} \ldots x_{j-1} y_1 y_2 x_1$, if $i = l$, then $C_{m+3}(z) = x_1 x_2 \ldots x_a x_{l+1} \ldots x_{j-1} y_1 y_2 x_{a+1} \ldots x_l x_j \ldots x_m z x_1$. Thus, in all cases, we have a contradiction. We may, therefore, assume that (recall that $b = l+1$)

$$\mathcal{A}(\{x_{a+1}, x_{a+2}, \ldots, x_l\} \to \{x_{b+1}, x_{b+2}, \ldots, x_m\}) = \emptyset.$$

Combining this with (6), we obtain

$$\mathcal{A}(\{x_1, x_2, \ldots, x_l\} \to \{x_{b+1}, x_{b+2}, \ldots, x_m\}) = \emptyset. \tag{7}$$

Assume first that $d^-(z,E) \geq 1$. Then by Claim 1(i), $d^+(z,F) = 0$. This together with (3) and (7) implies that $\mathcal{A}(\{z, x_1, x_2, \dots, x_l\} \cup Y \to \{x_{l+2}, x_{l+3}, \dots, x_m\}) = \emptyset$. Assume second that $d^-(z,E) = 0$. Since $x_l z \notin \mathcal{A}(D)$, we obtain $\mathcal{A}(\{x_1, x_2, \dots, x_l\} \cup Y \to \{z, x_{l+2}, x_{l+3}, \dots, x_m\}) = \emptyset$. So, in both cases we have that the subdigraph $D - x_{l+1}$ is not strong, which contradicts that D is 2-strong.

Subcase 3.2. $b = l + 1 = m$.

Assume that $a \geq 2$. As mentioned above, either $x_1 \to x_{a+1}$ or $x_{a+1} \to x_1$. Therefore, $C_{m+3}(z) = x_1 x_{a+1} \dots x_{m-1} y_1 y_2 x_2 \dots x_a x_m z x_1$ or $C_{m+2}(z) = x_1 \dots x_a x_m z x_{a+3} \dots x_{m-1} y_1 y_2$ $x_{a+1} x_1$. So, in both cases, we have a contradiction.

Assume next that $a = 1$. Then from $d^-(z, \{x_2, x_3, \dots, x_{m-1}\}) = 0$ (by Claims 2(i) and 2(iv)) and $d^-(z) \geq 2$ it follows that $x_1 \to z$. We know that $z \to \{x_{a+3}, \dots, x_l\}$. Using this, it is not difficult to see that if $x_i \to x_m$ with $i \in [2, m-2]$, then for $i = 2$, $C_{m+2}(z) = x_1 x_2 x_m z x_4 \dots x_{m-1} y_1 y_2 x_1$, and for $i \in [3, m-2]$, $C_{m+3}(z) = x_1 x_2 \dots x_i x_m z x_{i+1} \dots x_{m-1} y_1$ $y_2 x_1$, a contradiction. We may, therefore, assume that

$$d^-(x_m, \{x_2, x_3, \dots, x_{m-2}\}) = 0. \tag{8}$$

Now we consider the vertex x_1. If $x_j \to x_1$ with $j \in [2, m-2]$, then for $j = 2$, $C_{m+2}(z) = x_1 x_m z x_4 \dots x_{m-1} y_1 y_2 x_2 x_1$, and for $j \in [3, m-2]$, $C_{m+3}(z) = x_1 x_m z x_{j+1} \dots x_{m-1} y_1 y_2 x_2 \dots$ $x_j x_1$. Thus, in both cases, we have a contradiction. We may, therefore, assume that $d^-(x_1, \{x_2, x_3, \dots, x_{m-2}\}) = 0$. This together with (3), (8) and $d^-(z, \{x_2, x_3, \dots, x_{m-1}\}) = 0$ implies that

$$\mathcal{A}(\{x_2, x_3, \dots, x_{m-2}\} \to Y \cup \{z, x_1, x_m\}) = \emptyset.$$

This means that $D - x_{m-1}$ is not strong, which contradicts that D is 2-strong.

Case 4. $a = l - 2$. Taking into account Case 2 and the digraph duality, we may assume that $b \leq l + 2$.

Subcase 4.1. $a = l - 2$ and $b = l + 2$. Then by Claim 2, $d(z, \{x_{l-1}, x_l, x_{l+1}\}) = 0$. This together with (5) implies that

$$m - 1 \leq d(z, \{x_1, x_2, \dots, x_a, x_b, x_{b+1}, \dots, x_m\}) \leq a + m - b + 2$$

$$= m + l - 2 - l - 2 + 2 = m - 2,$$

a contradiction.

Subcase 4.2. $a = l - 2$ and $b = l + 1$. Then by Claim 2, $d(z, \{x_{l-1}, x_l\}) = 0$.

Assume first that $m \geq l + 2$. If there exist $i \in [l-1, l]$ and $j \in [l+2, m]$ such that $x_i \to x_j$, then $C(z) = x_1 x_2 \dots x_{l-2} x_{l+1} \dots x_{j-1} R(y_1, y_2) x_i x_j \dots x_m z x_1$ is a cycle of length at least $m + 2$, a contradiction. We may, therefore, assume that $\mathcal{A}(\{x_{l-1}, x_l\} \to \{x_{l+2}, x_{l+3}, \dots, x_m\}) = \emptyset$. This together with (3), the minimality of a and the maximality of b implies that $\mathcal{A}(\{x_1, x_2, \dots, x_l\} \to \{x_{l+2}, x_{l+3}, \dots, x_m\}) = \emptyset$. Therefore, if $d^-(z,E) = 0$, then $\mathcal{A}(\{x_1, x_2, \dots, x_l\} \cup Y \to \{z, x_{l+2}, x_{l+3}, \dots, x_m\}) = \emptyset$, and if $d^-(z,E) \geq 1$, then $d^+(z,F) = 0$ (Claim 1(i)) and $\mathcal{A}(\{z, x_1, x_2, \dots, x_l\} \cup Y \to \{x_{l+2}, x_{l+3}, \dots, x_m\}) = \emptyset$. Thus, in both cases, we have that $D - x_{l+1}$ is not strong, a contradiction.

Assume next that $m = l + 1$. Then $a = l - 2 = m - 3$. Let $a \geq 2$. From the minimality of a it follows that $d^-(x_m, \{x_1, x_2, \dots, x_{a-1}\}) = 0$. If there exist $i \in [1, a-1]$ and $j \in [a+1, a+2]$ such that $x_i \to x_j$, then it is easy to see that $C(z) = x_1 x_2 \dots x_i x_j \dots x_{m-1} R(y_1, y_2) x_{i+1} \dots x_a x_m z x_1$ is a cycle of length at least $m+2$, a contradiction. We may, therefore, assume that $\mathcal{A}(\{x_1, x_2, \dots, x_{a-1}\} \to \{x_{a+1}, x_{a+2}, x_{a+3} = x_m\}) = \emptyset$.

From this we have: if $d^-(z, \{x_1, x_2, \dots, x_{a-1}) = 0$, then

$$\mathcal{A}(\{x_1, x_2, \dots, x_{a-1}\} \to Y \cup \{z, x_{a+1}, x_{a+2}, x_{a+3}\}) = \emptyset,$$

if $d^-(z, \{x_1, x_2, \dots, x_{a-1}) \geq 1$, then by Claim 1(i), $zx_m \notin \mathcal{A}(D)$ and

$$\mathcal{A}(\{x_1, x_2, \dots, x_{a-1}\} \cup \{z\} \to Y \cup \{x_{a+1}, x_{a+2}, x_{a+3}\}) = \emptyset.$$

So, in both cases, we have that $D - x_a$ is not strong, which contradicts that D is 2-strong. Let now $a = 1$. Then $m = 4 = b = l + 1$ and $d(z, \{x_2, x_3\}) = 0$. This together with $d(z, Y) = 0$, $d^+(z) \geq 2$ and $d^-(z) \geq 2$ implies that $x_1 \to z \to x_4$, which contradicts Claim 1(i).

Case 5. $a = l - 1$. Taking into account Cases 3 and 4, we may assume that $b = l + 1$. Then $d(z, \{x_l\}) = 0$, and from (3), the minimality of a and the maximality of b it follows that

$$\begin{aligned} &\mathcal{A}(\{x_1, x_2, \dots, x_{l-1}\} \to Y \cup \{x_{l+2}, x_{l+3}, \dots, x_m\}) \\ &= \mathcal{A}(\{x_1, x_2, \dots, x_{l-2}\} \to Y \cup \{x_{l+1}, x_{l+2}, \dots, x_m\}) = \emptyset. \end{aligned} \tag{9}$$

It is not difficult see that: if $x_l \to x_j$ with $j \in [l+2, m]$, then $C(z) = x_1 x_2 \dots x_{l-1} x_{l+1} \dots x_{j-1} R(y_1, y_2) x_l x_j \dots x_m z x_1$ is a cycle of length at least $m + 3$, if $x_i \to x_l$ with $i \in [1, l-2]$, then $C(z) = x_1 x_2 \dots x_i x_l R(y_1, y_2) x_{i+1} \dots x_{l-1} x_{l+1} \dots x_m z x_1$ is a cycle of length at least $m + 3$. So, in both cases we have a contradiction. We may, therefore, assume that $d^+(x_l, \{x_{l+2} x_{l+3}, \dots, x_m\}) = d^-(x_l, \{x_1, \dots, x_{l-2}\}) = 0$. Then by (9),

$$\begin{aligned} &\mathcal{A}(\{x_1, x_2, \dots, x_{l-2}\} \to \{x_l, x_{l+1}, \dots, x_m\}) \\ &= \mathcal{A}(\{x_1, x_2, \dots, x_l\} \to \{x_{l+2}, x_{l+3}, \dots, x_m\}) = \emptyset. \end{aligned} \tag{10}$$

Assume that $m \geq l + 2$. If $d^-(z, E) \geq 1$, then $d^+(z, F) = 0$ (Claim 1(i)). This together with (3), (10), $d(z, \{x_l\}) = 0$ and $d(z, Y) = 0$ implies that $\mathcal{A}(\{z, x_1, x_2, \dots, x_l\} \cup Y \to \{x_{l+2}, x_{l+3}, \dots, x_m\}) = \emptyset$, which in turn implies that $D - x_{l+1}$ is not strong, a contradiction. We may, therefore, assume that $d^-(z, E) = 0$. Now it is not difficult to see that

$$\mathcal{A}(\{x_1, x_2, \dots, x_l\} \cup Y \to \{z, x_{l+2}, x_{l+3}, \dots, x_m\}) = \emptyset.$$

This means that $D - x_{l+1}$ is not strong, a contradiction.

Assume now that $m = l + 1$. By the digraph duality, we may assume that $a = l - 1 = 1$. Hence, $b = l + 1 = m = 3$. Then, since $d^+(z) \geq 2$ and $d^-(z) \geq 2$, $x_1 \to z \to x_m$, which contradicts Claim 1(i). The discussion of Case 5 is completed. Lemma 5 is proved. □

Now we are ready to prove the main result. For the convenience of the reader, we restate it here.

Theorem 9: *Let D be a 2-strong digraph of order $n \geq 8$ and z be a fixed vertex in $\mathcal{V}(D)$. Suppose that for any vertex $x \in \mathcal{V}(D) \setminus \{z\}$, $d(x) \geq n$, $d(z) \geq n - 4$, and D contains a cycle of length $n - 2$ passing through z. Then D is Hamiltonian.*

Proof. Suppose, on the contrary, that D contains a cycle $C_{n-2}(z) := x_1 x_2 \dots x_{n-2} x_1$ but it is not Hamiltonian. By Theorem 3 (or by Theorem 2), $d(z) \leq n - 2$. Let $\{y_1, y_2\} = \mathcal{V}(D) \setminus \mathcal{V}(C_{n-2}(z))$. Since $z \in \mathcal{V}(C_{n-2}(z))$, we have that $d(y_i) \geq n$. Using Lemma 1, it is easy to show that D contains no $C_{n-1}(z)$, $d(y_1) = d(y_2) = n$, $d(y_1, \mathcal{V}(C_{n-2}(z))) =$

$d(y_2, \mathcal{V}(C_{n-2}(z))) = n-2$ and $y_1 \leftrightarrow y_2$. If y_1 or y_2 is adjacent to every vertex x_i, $i \in [1, n-2]$, then D contains a cycle $C(z)$ of length at least $n-1$, a contradiction. We may, therefore, assume that y_1 and some vertex of $C_{n-2}(z)$ are not adjacent, say x_{n-2}. Then $d(y_1, \{x_1, x_2, \ldots, x_{n-3}\}) = n-2$. Since y_1 cannot be inserted into $x_1x_2\ldots x_{n-3}$, using Lemma 2, we obtain that $x_{n-3} \to y_1 \to x_1$. This together with $y_1 \leftrightarrow y_2$ implies that $d(x_{n-2}, \{y_1, y_2\}) = 0$ (for otherwise, D contains a cycle of length at least $n-1$ through z, which is a contradiction). Therefore, $d(y_2, \{x_1, x_2, \ldots, x_{n-3}\}) = n-2$, and by Lemma 2, $x_{n-3} \to y_2 \to x_1$. Then $C_{n-1} = x_1x_2\ldots x_{n-3}y_1y_2x_1$ is a cycle of length $n-1$. We know that C_{n-1} does not contain the vertex z. Therefore, $z = x_{n-2}$. Thus, we have that the conditions of Lemma 5 hold. Therefore, $d(z) \leq n-5$, which contradicts that $d(z) \geq n-4$. The theorem is proved. $\square$

In [15], Overbeck-Larisch proved the following sufficient condition for a digraph to be Hamiltonian-connected.

Theorem 10: (Overbeck-Larisch [15]). *Let D be a 2-strong digraph of order $n \geq 3$ such that, for each two non-adjacent distinct vertices x, y we have $d(x)+d(y) \geq 2n+1$. Then for each two distinct vertices u, v with $d^+(u)+d^-(v) \geq n+1$ there is a Hamiltonian (u, v)-path.*

Let D be a digraph of order $n \geq 3$ and let u and v be two distinct vertices in $\mathcal{V}(D)$. Follows Overbeck-Larisch [15], we define a new digraph $H_D(u, v)$ as follows: $\mathcal{V}(H_D(u, v)) = \mathcal{V}(D - \{u, v\}) \cup \{z\}$ (z a new vertex) and $\mathcal{A}(H_D(u, v)) = \mathcal{A}(D - \{u, v\}) \cup \{zy \mid y \in N^+_{D-v}(u)\} \cup \{yz \mid y \in N^-_{D-u}(v)\}$.

Now, using Theorem 7, we will prove the following theorem, which is an analogue of the Overbeck-Larisch theorem.

Theorem 11: *Let D be a 3-strong digraph of order $n+1 \geq 10$ with minimum degree at least $n+2$. If for two distinct vertices u, v, $d^+_D(u)+d^-_D(v) \geq n-2$ or $d^+_D(u)+d^-_D(v) \geq n-4$ with $uv \notin \mathcal{A}(\mathcal{D})$, then there is a Hamiltonian (u, v)-path in D.*

Proof. Let D be a 3-strong digraph of order $n+1 \geq 10$ and let u, v be two distinct vertices in $\mathcal{V}(D)$. Suppose that D and u, v satisfy the degree conditions of the theorem. Now we consider the digraph $H := H_D(u, v)$ of order $n \geq 9$. By an easy computation, we obtain that the minimum degree of H is at least $n-4$, and H has $n-1$ vertices of degrees at least n. Moreover, we know that H is 2-strong (see [10]). Thus, the digraph H satisfies the conditions of Theorem 7. Therefore, H is Hamiltonian, which in turn implies that in D there is a Hamiltonian (u, v)-path. $\square$

5. Conclusion

For Hamiltonicity of a graph G (undirected graph), there are numerous sufficient conditions in terms of the number $k(G)$ of connectivity, where $k(G) \geq 3$ (recall that for a graph G to be Hamiltonian, $k(G) \geq 2$ is a necessary condition) and the minimum degree $\delta(G)$ (or the sum of degrees of some vertices with certain properties), see the survey papers by Gould, e.g. [16]. This is not the case for the general digraphs. In [17], the author proved that: For every pair of integers $k \geq 2$ and $n \geq 4k+1$ (respectively, $n = 4k+1$), there exists a k-strong $(n-1)$-regular (respectively, with minimum degree at least $n-1$ and with minimum semi-degrees at least $2k-1 = (n-3)/2$) a non-Hamiltonian digraph of order n. In [1] (Page

253), it was showed that there is no k such that every k-strong multipartite tournament with a cycle factor has Hamiltonian cycle.

Based on the evidence from Theorem 9, we raise the following conjecture, the truth of which in the case $k = 0$ follows from Theorem 9.

Conjecture 2: *Let D be a 2-strong digraph of order n and z be a fixed vertex in $\mathcal{V}(\mathcal{D})$. Suppose that for any vertex $x \in \mathcal{V}(D) \setminus \{z\}$, $d(x) \geq n+k$ and $d(z) \geq n-k-4$, where $k \geq 0$ is an integer. Then D is Hamiltonian.*

References

[1] J. Bang-Jensen and G. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer, 2000.

[2] J.-C. Bermond and C. Thomassen, "Cycles in digraphs – A survey", *Journal of Graph Theory*, vol. 5, no. 1, pp. 1-43, 1981.

[3] D. Kühn and D. Osthus, "A survey on Hamilton cycles in directed graphs", *European Journal of Combinatorics*, vol. 33, pp. 750-766, 2012.

[4] A. Ghouila-Houri, "Une condition suffisante d'existence d'un circuit hamiltonien", *Comptes Rendus de I'Academie des Sciences Paris*, ser. A-B 251, pp. 495-497, 1960.

[5] M. Meyniel, "Une condition suffisante d'existence d'un circuit hamiltonien dans un graphe oriente", *Journal of Combinatorial Theory*, Ser. B, vol. 14, pp. 137-147, 1973.

[6] C.St.J.A. Nash-Williams, "Hamilton circuits in graphs and digraphs", *The many facets of graph theory, Springer Verlag Lecture Notes 110*, (Springer Verlag) pp. 237-243, 1969.

[7] C. Thomassen, "Long cycles in digraphs", *Proceedings of London Mathematical Society*, vol. 42, no. 3, pp. 231-251, 1981.

[8] S.Kh. Darbinyan, "Cycles of any length in digraph with large semi-degrees", *Aakdemy Nauk Armyan SSR Doklady*, (arXiv.1911.05998v1) vol. 75, no. 4, pp. 147-152, 1982 .

[9] C. Thomassen, "Long cycles in digraphs with constraints on degrees, Survey in Combinatorics", *Proc. 7th British Combinatorial Conf., London Math. Soc. Lecture Notes*, Cambridge University Press, vol. 38, pp. 211-228, 1979.

[10] S.Kh. Darbinyan, "Hamiltonian and strongly Hamilton-Connected digraphs", *Aakdemy Nauk Armyan SSR Doklady*, (arXiv.1801.05166v1), vol. 91, no. 1, pp. 3-8, 1990.

[11] M.K. Goldberg, L.P. Levitskaya and L.M. Satanovskiy, "On one strengthening of the Ghouila-Houri theorem", *Vichislitelnaya Matematika i Vichislitelnaya Teknika*, vol. 2, pp. 56-61, 1971.

[12] S.Kh. Darbinyan, "A sufficient condition for a digraph to be Hamiltonian", *Aakdemy Nauk Armyan SSR Doklady*, vol. 91, no. 2, pp. 57-59, 1990.

[13] R. Häggkvist and C. Thomassen, "On pancyclic digraphs", *Journal of Combinatorial Theory*, Ser. B, vol. 20, no. 1, pp. 20-40, 1976.

[14] J.A. Bondy and C. Thomassen, "A short proof of Meyniel's theorem", *Discrete Mathematics*, vol. 19, pp. 195-197, 1977.

[15] M. Overbeck-Larisch, "Hamiltonian pats in oriented graphs", *Journal of Combinatorial Theory*, Ser. B, vol. 21, pp. 76-80, 1976.

[16] R.J. Gould, “Resent Advances on the Hamiltonian Problem: Survey III”, *Graphs and Combinatorics*, vol. 30, pp. 1-46, 2014.

[17] S.Kh. Darbinyan, “Disproof of a conjecture of Thomassen ”, *Aakdemy Nauk Armyan SSR Doklady*, vol. 76, no. 2, pp. 51-54, 1983.

Գուիիլա-Հուրիի թեորեմի մի ընդլայնման մասին

Սամվել Խ. Դարբինյան

ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ
e-mail: samdarbin@iiap.sci.am

Ամփոփում

Ներկա աշխատանքում ապացուցվել է հետևյալ թեորեմը:

Թեորեմ: Դիցուք D-ն 2-ուժեղ կապակցված n-գագաթանի ($n \geq 8$) կողմնորոշված գրաֆ է, որի $n-1$ գագաթների աստիճանները փոքր չեն n թվից, իսկ z գագաթի աստիճանը փոքր չէ $n-4$ թվից: Եթե D-ն ն պարունակում է $n-2$ երկարությամբ ցիկլ, որը անցնում է z գագաթով, ապա D պարունակում է համիլտոնյան ցիկլ:

Բանալի բառեր՝ հակադարձ մատրից, երեքանկյունագծային մատրից, հերմիտյան մատրից, տյոպլիցյան մատրից:

Об одном расширении теоремы Гуйя-Ури

Самвел Х. Дарбинян

Институт проблем информатики и автоматизации НАН РА
e-mail: samdarbin@iiap.sci.am

Аннотация

В настоящей работе доказана следующая теорема. Теорема. Пусть D есть 2-сильно связный $n \geq 8$ вершинный орграф, в котором $n-1$ вершин имеют степень не меньше чем n , а вершина z имеет степень не меньше чем $n-4$. Если D содержит контур длины $n-2$, которий содержит вершину z, то D содержит гамильтонов контур.

Ключевые слова: орграф, гамильтонов контур, 2-сильно, гамильтоново-связный.

Mathematical Problems of Computer Science 58, 32–41, 2022.
doi:10.51408/1963-0090

UDC 519.218

Comparison of Model-Free Algorithms For Clustering GARCH Processes

Garik L. Adamyan

Yerevan State University
e-mail: garik.adamyan@ysu.am

Abstract

In this paper, we evaluate several model-free algorithms for clustering time series datasets generated by GARCH processes. In extensive experiments, we generate synthetic datasets in different scenarios. Then, we compare K-Means (for Euclidian and dynamic time warping distance), K-Shape, and Kernel K-Means models with different clustering metrics. Several experiments show that the K-Means model with dynamic time warping distance archives comparably better results. However, the considered models have significant shortcomings in improving the clustering accuracy when the amount of information (the minimum length of the time series) increases, and in performing accurate clustering when data is unbalanced or clusters are overlapping.
Keywords: Time series clustering, GARCH process, dynamic time warping, K-Means, K-Shape.
Article info: Received 02 May 2022; received in revised form 20 July 2022; accepted 29 September 2022.

1. Introduction

Time series clustering has been used in diverse scientific disciplines to discover patterns and extract valuable information from complex and massive datasets. These algorithms have a wide range of applications in many research areas, for instance, in finance, biology, and robotics [1].

Time series clustering approaches can be classified as feature-based, shape-based, and model-based [1]. It is noteworthy that these methods are based on dissimilarity measures on time series data, according to which the time series data points are grouped by some clustering method (for instance, PAM).

In general, shape-based methods use linear and non-linear transformations to align time series samples and calculate dissimilarity measures on aligned samples. Additionally, shape-based algorithms process the time series data directly without making any statistical assumptions about the underlying data generating processes. On the contrary, model-based methods make statistical assumptions on time series generating processes. In general, model-based approaches assume that time series samples are generated from specific models (for instance, ARIMA [2], Mixtures of ARIMAs [3]). Time series samples are transformed into fitted models, and then a suitable distance and a clustering algorithm are applied to the estimated model parameters.

Although several benchmarking results on different real-world datasets for non-parametric clustering methods can be found in ([4], [5], [6]), the comparison of non-parametric clustering methods on time series data generated from GARCH processes is not well studied. In this paper, we are interested in non-parametric models evaluation of time series data generated from the well-known GARCH process, which is the actual choice for modeling the volatility of returns on financial assets. We simulate multiple GARCH models with different data generating scenarios and compare several non-parametric time series clustering models.

Motivated by [4], for comparison we choose well-known partition-based time series clustering models: K-Means, K-Means with dynamic time warping and DTW barycenter averaging, K-Shape and Kernel K-Means models. Furthermore, we can find open-source implementations of these algorithms [7].

Although the main focus in the field of time series clustering comparison remains clustering accuracy metrics, in this work we also explore a number of other challenges of model-free methods. In particular, we study the ability of the above-mentioned model-free methods to cluster GARCH processes with imbalanced, overlapping clusters and also examine the impact of increasing information on clustering accuracy.

2. Related Work

In time series analysis research, benchmarking and numerical comparison have been recognized as integral steps to justify theoretical results. The importance of numerical comparison is emphasized in [8], where the authors reimplemented many time-series classification algorithms and compared them in 50 real-world datasets. The authors note that most reported methods have insignificant improvements regarding the variance of the evaluation metrics. This empirical evidence reclaimed the statement of the importance of the time series benchmark datasets and the empirical evaluation of the suggested methods.

Among the works that compare time series clustering models based on real-world datasets, we can mention ([4], [5], [6]) works. In [4], authors compare several partition, density, and hierarchical clustering methods to cluster all time series datasets available in the University of California Riverside (UCR) archive [9]. They conclude that the overall performance of the eight compared algorithms is quite similar with high dependence on the evaluation dataset.

The method of comparing time series clustering algorithms with synthetic, generated datasets also attracts a lot of attention among scholars. In addition to the actual clusters being known, this comparison method gives additional flexibility to examining the behavior of algorithms in different situations. In particular, scholars discussed the difference between stationary and non-stationary time series [10], the presence of noise in time series samples [11], the presence of noise clusters in time series dataset [11].

3. Clusters of GARCH

The GARCH process is introduced in [12] for statistical modeling of the volatility of returns on financial assets. The GARCH model has many extensions such as asymmetric GARCH [13], threshold GARCH [14]. The GARCH(p,q) model is defined as follows:

$$y_t = \mu_t + \epsilon_t$$

$$\epsilon_t = \sigma_t e_t, \quad \text{where} \quad e_t \quad i.i.d \quad E(e_t) = 0, var(e_t) = 1,$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2,$$

where

$$\omega > 0,$$
$$\alpha_i \geq 0, i = 1, 2, ..., p,$$
$$\beta_j \geq 0, i = 1, 2, ..., q.$$

The GARCH(p, q) model admits a strictly stationary solution with a finite variance if and only if

$$\sum_{i=1}^{p} \alpha_i + \sum_{j=1}^{q} \beta_j < 1. \tag{1}$$

Moreover, this strictly stationary solution is also unique. [15]

For the evaluation of non-parametric models, we chose constant zero mean specification for the GARCH model because it is advised to standardize input data prior to clustering. In addition, we choose the innovations e_t as standard Gaussian innovations. So $\mu_t = 0$ and $e_t \sim \mathcal{N}(0,1)$.

In order to measure the clustering accuracy, we need to define the ground truth clusters of GARCH processes. Let $N, K, T \in \mathbb{N}$ where K is the number of clusters, N is the number of samples and T is the time sample size of each series. In this paper, we consider samples with a fixed time size T, because some of the models (ex. KM-E) support samples with fixed length. We denote by $P^i = (\omega, \alpha_1, \alpha_2, ..., \alpha_{p_i}, \beta_1, \beta_2, ..., \beta_{q_i})$ the vector of all parameters for the given GARCH(p_i,q_i) model.

Let $\{P^i\}_{i=1}^K$ be a family of GARCH process parameters, where K is a number of clusters. Assume that each $P^i (i = 1, 2, ..., K)$ is unique and all the parameters satisfy (1) in order to provide a strict stationary solution of the corresponding model. We are given N samples of time series $Y_i = \{y_t^i\}_{i=1}^T$, where each sample is generated from one of the K GARCH processes.

Definition 1. *We say that Y_i and Y_j samples are from the same cluster if they are generated from the same GARCH process.*

In other words, a cluster of GARCH processes is a set of samples that are generated with the same parameters. The uniqueness of the parameters P^i and Definition 1 imply that the given sample belongs to exactly one cluster.

4. Evaluation Models

For evaluation, we choose well-known non-parametric time series clustering models such as K-Means with Euclidean (KM-E) and dynamic time warping metrics (KM-DTW), K-Shape, and Kernel K-Means with Fast Global Alignment Kernel (KKM-GAK) models. KM-E uses Euclidean distance, for cluster assignment and means averaging for the barycenter (centroid) computation. It is known that the Euclidean distance metric is not the most accurate metric for measuring time series similarities. Firstly, to use Euclidean distance, we need to take into account the order of elements in the time series; secondly, the Euclidean distance does not consider a phase shift between two curves or a length difference between the series. In this paper, we consider this model for comparison with more complex approaches.

KM-DTW uses dynamic time warping [16] for cluster assignment and DTW barycenter averaging (DBA)[17] algorithm for averaging time series within the same cluster.

k-Shape [18] is a partitional clustering algorithm that relies on an iterative refinement procedure similar to the one used in K-Means. To measure the distance between time series, K-Shape uses a normalized version of the cross-correlation measure to consider the shapes of time series while comparing them. During the iterative procedure, this model minimizes the sum of squared distances between the sequences of time series.

Kernel K-Means[19] is an alternative clustering algorithm that uses kernel functions as a nonlinear mapping from the input space to a higher dimensional space. By using kernels, Kernel K-Means can separate clusters in higher dimensional space, even if the input data is not non-linearly separable in the input space. For treating time series data, practitioners usually used Global Alignment Kernels [20]. We will refer to this algorithm KKM-GAK.

The problem is to generate synthetic datasets and evaluate non-parametric models for clustering time series processes generated by the GARCH model.

5. Assessment Metrics

In practice, the use of clustering methods is due to working with unlabeled datasets. As a result, we can find evaluation metrics that can evaluate clustering models without having labeled data. These types of metrics are called internal. By the method of our data generating process, we can use external measures, which assume that ground truth labels are available. Examples of this type of metrics are the Rand Index (RI) [21], the Adjusted Rand Index (ARI) [22], the Adjusted Mutual Information (AMI)[23].

Following the evaluation made in [4] in our study, we choose the Adjusted Rand Index, because the values of this metric are consistently low for random cluster assignments and do not depend on the number of clusters.

6. Experiments

To evaluate non-parametric models, we simulate random datasets with different setups. In the first experiment, we measure the ability of the models of clustering different numbers of clusters. For this purpose, we generate datasets for 2, 4, 8, and 10 clusters, respectively. For each number of clusters, we generate random parameter families, which satisfy (1) for guaranteeing a unique and stationary solution of processes. For the purpose of generating a family of parameters, we constrain the maximum length of p and q by 5. This constraint is inherited from the common choice of GARCH models with fewer parameters. For every parameter vector P^i (cluster), we generate samples for the given cluster and separate them into training and testing parts (30% testing) and repeat this process for averaging purposes. In Table 1, we present the results of the first experiment evaluated with the AMI metric. We can see that the KM-DTW model outperforms other models. In the second experiment,

Table 1: AMI score for different N clusters

N clusters	KM-E	KM-DTW	k-Shape	KKM-GAK
2	0.003+-0.001	0.325+-0.403	0.004+-0.009	0.003+-0.002
4	0.004+-0.001	0.463+-0.129	0.02+-0.007	0.002+-0.001
6	0.018+-0.016	0.578+-0.151	0.043+-0.021	0.001+-0.0005
8	0.006+-0.003	0.498+-0.077	0.005+-0.011	0.001+-0.0005
10	0.005+-0.01	0.624+-0.03	0.062+-0.022	0.0001+-0.00005

we measure the clustering quality in scenarios when the amount of information increases. We generate datasets with 5 clusters and 100 samples in each cluster. We set $T = 1000$ and consider 5 intervals on the time axis. We train and evaluate models in the first interval and consequently add information. From the second experiment, we can see that the KM-DTW model outperforms other models, but we do not observe increased accuracy as a result of adding information. There is a significant increase in the accuracy of the KM-DTW model when the number of samples increases from 200 to 400, but further increases in the number of samples do not improve the accuracy of the model. The K-Shape model also shows a slight improvement in accuracy when the number of samples increases from 800 to 1000. Given that model-based methods rely on ML/Quasy ML estimates of the parameters of GARCH models and also the asymptotic properties of these estimates, this experiment may suggest that model-based methods have the potential to increase clustering accuracy as information increases. The results of the second experiment are displayed in Fig. 1.

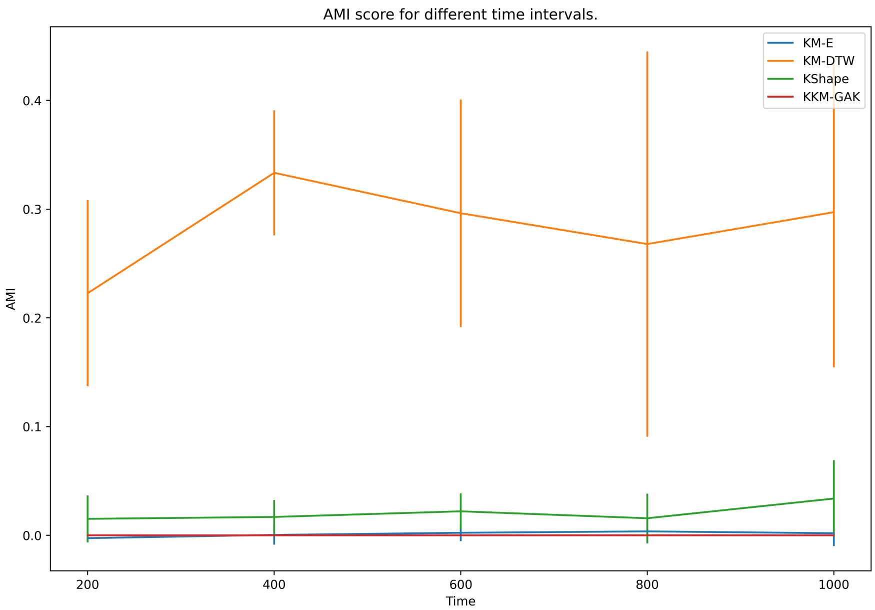


Fig. 1. AMI for different time intervals.

Fig. 2 shows the results of the third experiment. In this experiment, we measure the ability of the KM-DTW model to cluster an imbalanced dataset. For the fairness of the experiment, we generate time series samples with the GARCH(1,1) process and ensure that parameters satisfy (1). In addition, we constrain the L_2 norm of generated parameters to obtain non-overlapping clusters. We generate a dataset with different sample ratios and increase the ratio to 1. In the figure, we can observe that the best model for other experiments KM-DTW is dependent on cluster imbalance. This experiment shows that the claim made in [24] that centroid-based methods should be adapted to unbalanced scenarios also holds in the domain of time series clustering.

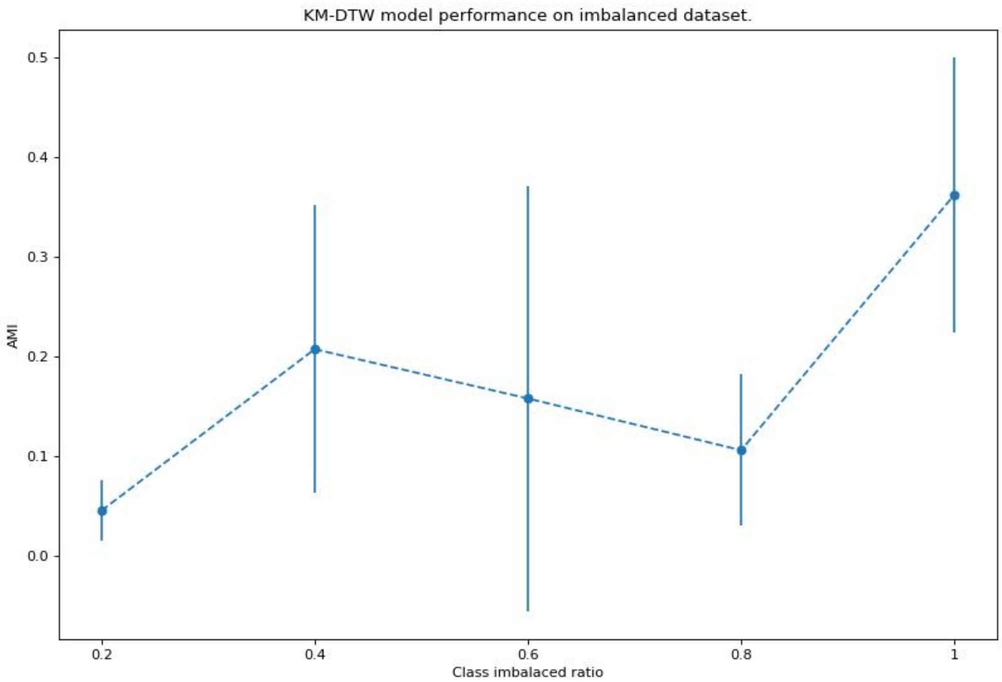


Fig. 2. Results for clustering imbalanced dataset.

Moreover, we measure the effect of the L_2 norm of generated parameters in clustering accuracy. We generate parameters for GARCH(1,1) process so that the parameters satisfy the current restriction on the L_2 norm. Throughout the experiment, we increase the bounds of the L_2 norm. During each step, we generate a balanced dataset with $T = 500$, $C = 2$, and 100 samples per cluster. We train models ten times for averaging purposes. We can observe that the KM-DTW model depends on clusters overlapping and increasing the bounds of parameters L_2 norm results in improvement of AMI. This problem is directly related to the ability of the similarity measure used in the KM-DTW algorithm to distinguish realizations of the GARCH process with parameters that are close to each other with the L2 norm.

7. Conclusion and Future Work

In this work, several non-parametric clustering algorithms for clustering time series datasets generated by GARCH processes are evaluated. We generate multiple datasets and conduct multiple experiments to evaluate the K-Means (with Euclidean and dynamic time warping distance), K-Shape, and Kernel K-Means models. In the first experiment, we evaluate the ability of models to cluster different numbers of clusters. The results of the first experiment

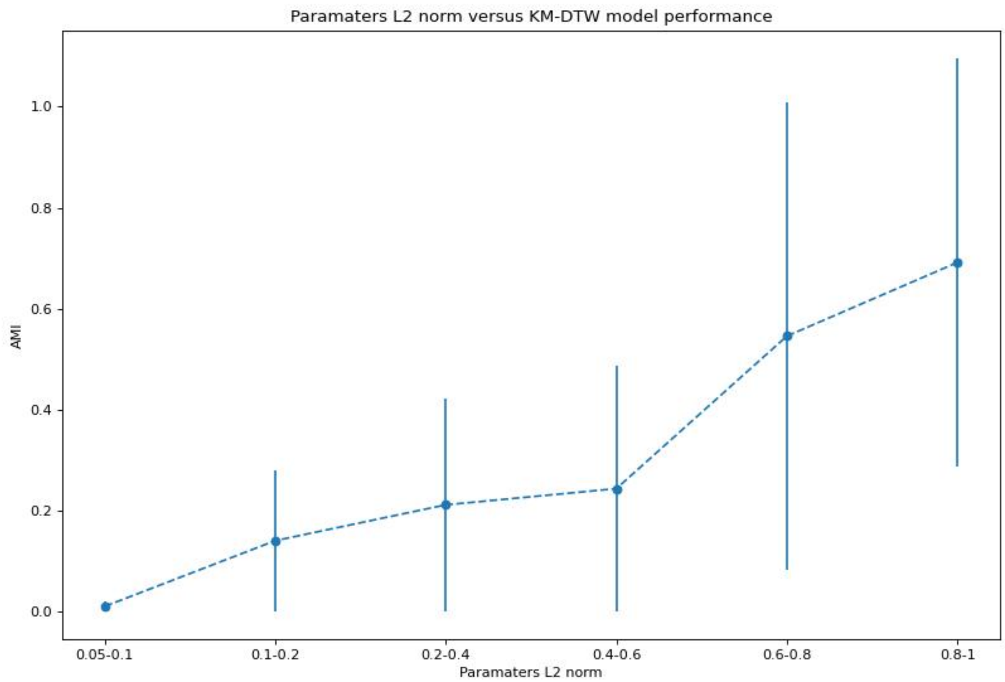


Fig. 3. GARCH parameters vector L_2 norm versus AMI score.

are displayed in Table 1. In the second experiment, we measure the clustering quality in the scenarios when the amount of information increases. We generate a dataset with 1000 time length and increase the information set. The results of the second experiment are shown in Fig. 1. During both experiments, the KM-DTW model shows better results. In the third experiment, we measure the ability of the KM-DTW model to cluster imbalanced datasets by generating multiple datasets with imbalanced samples in the cluster. The results are provided in Fig. 2. In the fourth experiment, we measure the ability of the KM-DTW model to cluster overlapping clusters. We constrain the norm of the parameters of the GARCH(1,1) model and evaluate the KM-DTW model. The experiment shows that KM-DTW is highly dependent on the norm of the generated parameters. The results of the fourth experiment are shown in Fig. 3.

We hope that our findings can motivate scholars to examine the discussed issues related to clustering accuracy, cluster overlapping, and available information effect. We think that already designed GARCH-based clustering methods have the potential to overcome these problems, so it is important to conduct similar experiments to show this. Moreover, as a direct application of our findings, it is worth applying clustering algorithms to the real-world financial dataset.

References

[1] S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah, “Time-series clustering a decade review,” *Information Systems*, vol. 53, p. 1638, 2015.

[2] M. Corduas and D. Piccolo, “Time series clustering and classification by the autoregressive metric,” *Computational Statistics amp; Data Analysis*, vol. 52, no. 4, p. 18601872, 2008.

[3] Y. Xiong and D.-Y. Yeung, “Mixtures of arma models for model-based time series clustering,” *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*

[4] A. Javed, B. S. Lee, and D. M. Rizzo, “A benchmark study on time series clustering,” *Machine Learning with Applications*, vol. 1, p. 100001, 2020.

[5] N. Begum, L. Ulanova, J. Wang, and E. Keogh, “Accelerating dynamic time warping clustering with a novel admissible pruning strategy,” *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

[6] J. Paparrizos and L. Gravano, “Fast and accurate time-series clustering,” *ACM Transactions on Database Systems*, vol. 42, no. 2, p. 149, 2017.

[7] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar, and E. Woods, “Tslearn, a machine learning toolkit for time series data,” *Journal of Machine Learning Research*, vol. 21, no. 118, pp. 1–6, 2020.

[8] E. Keogh and S. Kasetty *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 349–371, 2003.

[9] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, “The ucr time series classification archive,” July 2015. www.cs.ucr.edu/ eamonn/time$_series_data/$.

[10] S. P. Daz and J. A. Vilar, “Comparing several parametric and nonparametric approaches to time series clustering: A simulation study,” *Journal of Classification*, vol. 27, no. 3, pp. 333–362, 2010.

[11] P. DUrso, L. De Giovanni, R. Massari, and D. Di Lallo, “Noise fuzzy clustering of time series by autoregressive metric,” *METRON*, vol. 71, no. 3, p. 217243, 2013.

[12] T. Bollerslev, “Generalized autoregressive conditional heteroskedasticity,” *Journal of Econometrics*, vol. 31, no. 3, p. 307327, 1986.

[13] L. Hentschel, “All in the family nesting symmetric and asymmetric garch models,” *Journal of Financial Economics*, vol. 39, no. 1, p. 71104, 1995.

[14] J. Park, J. Baek, and S. Hwang, “Persistent-threshold-garch processes: Model and application,” *Statistics amp; Probability Letters*, vol. 79, no. 7, p. 907914, 2009.

[15] A. M. Lindner, “Stationarity, mixing, distributional properties and moments of garch(p, q)processes,” *Handbook of Financial Time Series*, p. 4369, 2009.

[16] *Dynamic Time Warping*, pp. 69–84. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[17] F. Petitjean, A. Ketterlin, and P. Ganarski, “A global averaging method for dynamic time warping, with applications to clustering,” *Pattern Recognition*, vol. 44, no. 3, pp. 678–693, 2011.

[18] J. Paparrizos and L. Gravano, “K-shape,” *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 2015.

[19] I. S. Dhillon, Y. Guan, and B. Kulis, “Kernel k-means,” *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’04*, 2004.

[20] M. Cuturi, “Fast global alignment kernels,” in *ICML*, 2011.

[21] L. Hubert and P. Arabie, “Comparing partitions,” *Journal of Classification*, vol. 2, no. 1, p. 193218, 1985.

[22] J. M. Santos and M. Embrechts, "On the use of the adjusted rand index as a metric for evaluating supervised classification," *Artificial Neural Networks ICANN 2009*, p. 175184, 2009.

[23] S. Romano, N. the Vinh, J. C. Bailey, and K. M. Verspoor, "Adjusting for chance clustering comparison measures," *Journal of Machine Learning (JMLR)*, pp. 4635–4666, vol. 17, no. 1, 2016.

[24] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, p. 221–232, 2016.

GARCH պրոցեսների կլաստերիզացիայի համար մոդելներից անկախ ալգորիթմների համեմատություն

Գարիկ Լ. Ադամյան

Երևանի պետական համալսարան, Երևան, Հայաստան
e-mail: garik.adamyan@ysu.am

Ամփոփում

Հոդվածում մենք գնահատում ենք մի քանի մոդելներից անկախ կլաստերիզացիայի ալգորիթմների GARCH պրոցեսներով գեներացված ժամանակային շարքերի տվյալների կլաստերավորման ունակությունը: Լայնածավալ փորձերի ընթացքում մենք գեներացնում ենք սինթետիկ տվյալների հավաքածուներ տարբեր սցենարներով: Այնուհետև, մենք համեմատում ենք K-Means մոդելները (Էվկլիդեսյան և ժամանակի դինամիկ փոխակերպման մետրիկաներով), K-Shape և Kernel K-Means մոդելների տարբեր կլաստերային չափիչներով: Մի քանի փորձերը ցույց են տալիս, որ K-Means մոդելը ժամանակի դինամիկ փոխակերպման մետրիկայով ցույց է տալիս համեմատաբար ավելի լավ արդյունքներ: Այնուամենայնիվ, դիտարկված մոդելներն ունեն զգալի թերություններ ինֆորմացիայի (ժամանակային շարքի նվազագույն երկարությունը) քանակի ավելացման հետ կլաստերավորման ճշգրտության բարձրացման հետ կապված, ինչպես նաև տվյալների անհավասարակշռության կամ կլաստերի համընկնման դեպքում ճշգրիտ կլաստերավորում իրականացնելու հարցում:

Բանալի բառեր՝ ժամանակային շարքերի կլաստերիզացիա, GARCH պրոցեսներ, ժամանակի դինամիկ փոխակերպում, K-Means, K-Shape.

Сравнение безмодельных алгоритмов кластеризации GARCH-процессов

Гарик Л. Адамян

Ереванский государственный университет, Ереван, Армения
e-mail: garik.adamyan@ysu.am

Аннотация

В этой статье мы оцениваем некоторые безмодельные алгоритмы кластеризации наборов данных временных рядов, сгенерированных GARCH процессами. В обширных экспериментах мы генерируем синтетические наборы данных для различных сценариях. Затем мы сравниваем модели K-Means (с метриками евклидовой и динамической трансформации временной шкалы), модели K-Shape и Kernel K-Means с различными метриками кластеризации. Несколько экспериментов показывают, что модель K-Means с метрикой динамической трансформации временной шкалы дает сравнительно лучшие результаты. Однако рассмотренные модели имеют существенные недостатки в повышении точности кластеризации при увеличении количества информации (минимальной длины временного ряда), а также при несбалансированности данных или перекрытии кластеров.

Ключевые слова: кластеризация временных рядов, процесс GARCH, динамическая деформация времени, K-Means, K-Shape.

Mathematical Problems of Computer Science 58, 42–51, 2022.
doi:10.51408/1963-0091

UDC 519.6, 004.9

A Brief Comparison Between White Box, Targeted Adversarial Attacks in Deep Neural Networks

Grigor V. Bezirganyan and Henrik T. Sergoyan

Department of Mathematics, Technical University of Munich, Muinch, German
e-mail: grigor.bezirganyan@tum.de, henrik.sergoyan@tum.de

Abstract

Today, neural networks are used in various domains, in most of which it is critical to have reliable and correct output. This is why adversarial attacks make deep neural networks less reliable to be used in safety-critical areas. Hence, it is important to study the potential attack methods to be able to develop much more robust networks. In this paper, we review four white box, targeted adversarial attacks, and compare them in terms of their misclassification rate, targeted misclassification rate, attack duration, and imperceptibility. Our goal is to find the attack(s), which would be efficient, generate adversarial samples with small perturbations, and be undetectable to the human eye.
Keywords: Adversarial Attacks, Robustness, Machine Learning, Deep Learning.
Article info: Received 26 April 2022; received in revised form 4 July 2022; accepted 29 July 2022.

1. Introduction

Nowadays, deep neural networks are becoming more and more popular to solve problems in various domains, including safety-critical areas such as medicine, self-driving cars, etc. Unfortunately, techniques to fool deep learning models have recently come out to provide incorrect outputs [1]. Particularly, in the image classification domain, an attacker can create an altered image, which will be misclassified by a model but will be classified correctly by a human. This altered image is often referred to as an *adversarial example*, and this process as an *adversarial attack*. To be protected against such attacks, researchers try to create methods to make the models more robust against such perturbations. Studying adversarial attacks and their potential helps us develop better countermeasures against them.

In this paper, we will discuss some of the adversarial algorithms and test them against an image classification model. We then compare the results of the experiments in terms of their misclassification rate, targeted misclassification rate, attack duration, and imperceptibility.

1.1 Definitions and Notations

1.1.1 Poisoning Attacks vs Evasion Attacks

In **poisoning attacks**, the attacker tries to insert fake samples (i.e., data samples with wrong labels) into the training dataset, which will make the model learn on those fake samples and output wrong results. This kind of attack is possible when the attacker has the means to import those fake samples into the training set. In contrast, in **evasion attacks**, the attacker does not need access to the dataset. In this case, the attacker creates adversarial samples, which are similar and hard to distinguish by a human from the original samples but are misclassified by the trained model.

1.1.2 Attacker's Knowledge of the Model

Based on how much information the attacker has about the model, attacks can be classified into white-box, black-box, and gray-box attacks. In the **white box** scenario, the attacker has full knowledge about the model architecture and uses this knowledge to generate adversarial examples. In contrast, in the **black-box** setup, the attacker does not know the architecture. Instead, the attacker observes the output of the model from the given input. Some of the attacks assume access to the soft labels (i.e., probability or likelihood score of belonging to a class), while others try to generate examples based on only hard labels (i.e., class labels without the score). In the **gray-box** setting, the attacker has an access to the original model and trains a generative model on it. When the generative model is ready, the attacker uses that model to generate adversarial samples. Hence, the original model is no more needed. Recently, in [2] another category was introduced, called **no-box** attacks. In contrast to black-box attacks, the attacker cannot query the model, instead, he has a small number of samples from the same domain as the victim. The authors train an auto-encoder on those samples and then generate the adversarial examples using the features learned from the auto-encoder.

1.1.3 Targeted vs Non-Targeted Attacks

In the **targeted attack**, the attacker tries to misclassify the given sample into a specific target label. In contrast, in **non-targeted** attacks, the attacker tries to classify the sample into any other class.

1.2 Our Goal and Contribution

In this paper, we try to overview some of the adversarial attack techniques and, running experiments in the same setting, compare them based on:

- **Misclassification:** What percentage of the adversarial samples were misclassified
- **Targeted Misclassification**: What percentage of the adversarial samples were successfully misclassified to the target class
- **Imperceptibility:** How much the adversarial example looks like the original image
- **Duration of the attack**: How long it takes to generate an adversarial example

In this paper, we will concentrate only on white-box and target attacks. In particular, we will discuss and experiment with the Fast Gradient Sign Method [1], Projected Gradient Descent [3], AutoPGD [4], and FW + Dual LMO [5]. We chose these attack methods as FGSM is one of the first and simplest methods, which is still popular today. PGD is the most popular, as even many new state-of-the-art methods are modified versions of the PGD attack. AutoPGD, being one of those variations, achieves state-of-the-art results according to the authors. And while these attacks use ℓ_p norms, we also chose FW + Dual LMO as an example of an attack that uses another norm (Wasserstain norm in this case).

2. Attack Mechanisms

In this section, we will briefly overview the attacks, which will be used for experimentation further in the paper.

In our attacks, we are given a set of input images $x \in \mathbb{R}^{n\times n}$, and our goal is to craft an adversarial example $x' \in \mathbb{R}^{n\times n}$ that will be misclassified by the deep learning model $F : \mathbb{R}^{n\times n} \to \mathbb{N}$. Since we are discussing targeted attacks, we want to misclassify the adversarial sample into our desired target class $t \in \mathbb{N}$ instead of the original class $y \in \mathbb{N}$. Furthermore, the perturbation we add to the image should be as small as possible, not to be detected by a human. So, we can formulate the problem in the following way: Given a Neural Network $F : \mathbb{R}^{n\times n} \to \mathbb{N}$, input image $x \in \mathbb{R}^{n\times n}$ with a label $y \in \mathbb{N}$, a distance function $||\cdot||$ and a perturbation budget $\epsilon \in \mathbb{R}$ find an $x' \in \mathbb{R}$ such that

$$\begin{aligned} & F(x') = t \neq y \\ \text{s.t.} \quad & ||x' - x|| \leq \epsilon. \end{aligned} \tag{1}$$

In our case, the distance functions will be l_1, l_2, l_∞ distances or the Wasserstein distance.

2.1 Fast Gradient Sign Method (FGSM)

Since we can access the gradients of the network in the white-box setting, what most of the gradient-based attacks do, is to fix the network weight and maximize the loss by updating the image. For that, they add a small perturbation $\eta \in \mathbb{R}^{n\times n}$ to the original image:

$$x' = x + \eta$$

The most efficient way to maximize the loss would be to add noise in the same direction as the gradients. [1] introduced an attack method, where they do exactly that: add a perturbation in a direction that will increase the loss function $\mathcal{L}$ between the adversarial example and the original label

$$x' = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(\theta, x, y)). \tag{2}$$

We can see that in this way the maximum allowed perturbation is added, while still being in the ϵ ball.

For a targeted setting, the update step will become:

$$x' = x - \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(\theta, x, t))$$

in other words, a perturbation is added to minimize the loss between the adversarial sample and the target class t.

2.2 Projected Gradient Descent

The Projected Gradient Descent attack (PGD) or Basic Iterative Method (BIM) was introduced in [3], where they transformed the FGSM [1] one-step attack into an iterative one by performing the update step (2) multiple times with a small step size $\alpha \in \mathbb{R}^{n\times n}$. This will work better, as the FGSM adds the maximum allowed perturbation, but does not guarantee to maximize the loss within the allowed $\epsilon-$ball. In contrast, in an iterative approach, the algorithm is more likely to find the maxima. To ensure that the adversarial sample remains in the ϵ neighborhood, PGD projects the sample back to the ϵ ball after each update step. In other words, it performs projected gradient descent (or ascent) on the input sample. The update steps for targeted and untargeted attacks will be as follows:

$$x^{(i+1)} = \Pi_\epsilon(x^{(i)} + \alpha \cdot \text{sign}(\nabla_{x^{(i)}}\mathcal{L}(\theta, x^{(i)}, y))) \tag{3}$$

$$x^{(i+1)} = \Pi_\epsilon(x^{(i)} - \alpha \cdot \text{sign}(\nabla_{x^{(i)}}\mathcal{L}(\theta, x^{(i)}, t))) \tag{4}$$

So, the attacker tries to find a perturbation that either finds the maximum loss between x' and y (3) (untargeted attack), or the minimum loss between x' and t (4) (targeted attack).

2.3 Auto-Projected Gradient Descent

It has recently been suggested [4] that the Cross-Entropy loss and the fixed step size of the PGD attack [3] may be two reasons for its potential failure. They propose an alternative loss function and a new gradient-based method, Auto-PGD, which does not require a fixed step size.

They divide their method into two phases: an exploration phase and an exploitation phase. During the exploration phase, they search for good initial points, while in the exploitation phase, they try to maximize the accumulated knowledge. The step size value depends on the trend of optimization. If the objective function decreases rapidly, then the step size does not need to be changed, otherwise, if it decreases slowly, the step size is reduced.

2.4 Wasserstein Attack

The Wasserstein adversarial attack was introduced in [6]. Here they proposed to use the Wasserstein distance instead of the commonly used ℓ_p distances. For images, the Wasserstein distance can be seen as the cost of redistributing pixel mass. For example, while rotations change ℓ_p norms dramatically, they only slightly change the Wasserstein distance.

So, what their algorithm does, is to do a PGD attack [3], but instead of projecting on an ℓ_p norm, they project on the Wasserstein ball. However, since the projection onto the Wasserstein ball is computationally expensive, they make an approximation by performing modified Sinkhorn iterations [7].

[5] improved the algorithm by introducing an exact but still efficient projection operator. They also introduce an adversary generating method based on the Frank-Wolfe [8] method equipped with a suitable linear minimization oracle and show that it works very fast for Wasserstein constraints.

In this paper, we will use that Frank-Wolfe method (FW + Dual LMO) for the experiments.

3. Experiments

3.1 Goal

In this experiment, our goal is to run FGSM [1], PDG [3], AutoPGD [4], and FW + Dual LMO [5] attacks on the same environment and compare them in terms of misclassification, targeted misclassification, attack duration, and imperceptibility.

3.2 Setup

We are performing our experiments on a pre-trained ResNet-18 [9] classifier on the CIFAR-10 dataset [10], with initial 92.4% accuracy on the test set. We generate the adversarial examples on a server with an Nvidia GeForce GTX 1080-Ti GPU.

We use the Adversarial Robustness Toolkit (ART) [11] for FGSM [1] and PGD [3] and AutoPGD [4] attacks, and the original implementation by the authors for FW + Dual LMO [5]. We run each of the adversarial attacks with a set of epsilon values in $\epsilon \in (0, 0.5]$ and for all target classes. We use ℓ_p norms for FGSM, PGD, and AutoPGD, and we use the Wasserstein distance for the FW + Dual LMO. All the other hyper-parameters are left to their default values. For the FW + Dual LMO, in the original implementation, there was no option for targeted attacks. Hence, we modified their implementation and added the option for target attacks. For that we converted the problem:

$$\begin{aligned} &\text{maximize} \quad \mathcal{L}(F(x'), y) \\ &\text{subject to } ||x' - x|| \leq \epsilon \end{aligned}$$

to

$$\begin{aligned} &\text{minimize} \quad \mathcal{L}(F(x'), t) \\ &\text{subject to } ||x' - x|| \leq \epsilon \end{aligned}$$

We log the duration of the attack, the misclassification rate, and the targeted misclassification rate for later comparison. The source code for the experiment can be found $https://github.com/bezirganyan/adversarial_arenahere.$

4. Results

4.1 Targeted Misclassification and Misclassification Rate

We first look at the average misclassification and targeted misclassification scores that each of our models was able to achieve for some $\epsilon \in (0, 0.5]$. In Table 1, we can see average misclassification and targeted misclassification rates for the best epsilon of each attack. As we can see from the ℓ_p attacks, the ℓ_∞ norm yields the highest scores in our setup. Hence, from now on we will use the ℓ_∞ norm for further comparisons. Note that this does not mean that the ℓ_∞ norm is better since we could get similar scores and similar perturbations for higher ϵ values under other norms, as the ℓ_∞ attack will add a higher amount of perturbation under the same epsilon.

Furthermore, we can see that from the ℓ_p attacks in terms of targeted misclassification rate, the PGD, and AutoPGD attacks yield very high scores leaving the FGSM attack behind with a huge margin. In general, PGD and AutoPGD attacks behave almost identically in our experiments. We hypothesize that this is because we are testing on an undefended model, on which they both reach their maximum potential limit. The developers of the ART framework confirmed that on their tests on defended models in an untargeted setting, AutoPGD behaved slightly better. We, hence, plan to test and compare the models on a defended model in our future work. In Fig. 1, we can see the Misclassification and Targeted misclassification rates of the attacks for different epsilons and under the ℓ_∞ norm. We can see that in terms of misclassification and targeted misclassification rates the PGD and AutoPGD attack perform best within the ℓ_p attacks by having around 90% misclassification rate even for very small epsilon.

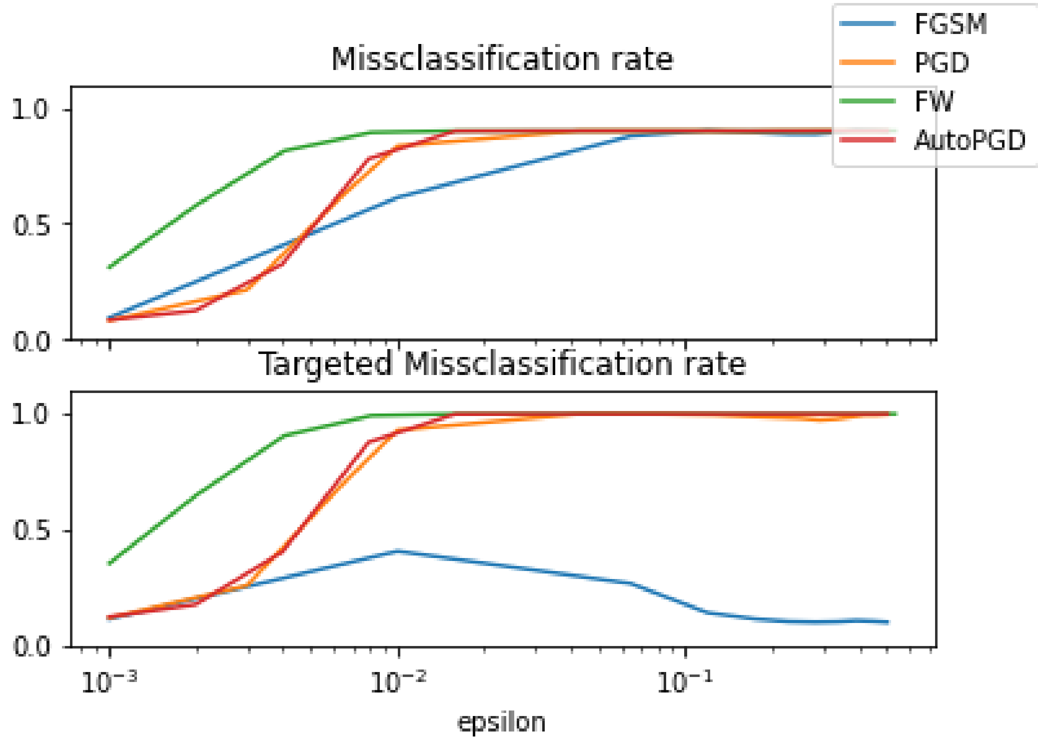


Fig. 1. Average misclassification and targeted misclassification rates for different ϵ values under ℓ_∞ and Wasserstein (FW) norms.

Furthermore, we can see that for the FGSM attack, the targeted misclassification does not increase monotonically. The reason for this can be that since the FGSM is not an iterative algorithm and performs just one step, it overshoots when the epsilon is too big and misses the target class.

The FW+Dual LMO attack performs best in terms of both misclassification and targeted misclassification rates. Nevertheless, we cannot compare the amount of perturbation under ℓ_∞ and Wasserstein norms, since they imply different amounts of changes to the image. Hence, we will need to combine these results with the visual ones to be able to make a fair comparison.

4.2 Duration

In Table 2, we can see the time duration needed to generate an adversarial example. Being a simple one-step attack, FGSM leads the competition followed by the PGD and AutoPGD

Table 1: Average misclassification and average targeted misclassification rates for different norms

attack	norm	miscl	targ. miscl.
AutoPGD	ℓ_1	0.0832	0.1100
PGD	ℓ_1	0.0810	0.1086
FGSM	ℓ_1	0.0879	0.1116
AutoPGD	ℓ_2	0.8968	0.9977
FGSM	ℓ_2	0.6157	0.3914
PGD	ℓ_2	0.8927	0.9925
AutoPGD	ℓ_∞	0.9000	1.0000
FGSM	ℓ_∞	0.9149	0.5515
PGD	ℓ_∞	0.9022	1.0000
FW	was	0.9000	1.0000

attacks. PGD, which performs much better than FGSM in terms of targeted misclassification rate, is around 71 times slower. The slowest is the FW + Dual LMO attack, which performs around 400 times slower than the FGSM attack.

4.3 Duration

In Table 2, we can see the time duration needed to generate an adversarial example. Being a simple one-step attack, FGSM leads the competition followed by the PGD and AutoPGD attacks. PGD, which performs much better than FGSM in terms of targeted misclassification rate, is around 71 times slower. The slowest is the FW + Dual LMO attack, which performs around 400 times slower than the FGSM attack.

Table 2: Duration of generating an adversarial example in seconds.

FGSM	PGD	AutoPGD	FW+Dual LMO
0.7	50	87	338

4.4 Imperceptibility

One of the most important aspects of Adversarial attacks is that they should be undetected by the human eye. Hence, in this section, we study how detectable are the adversarial samples generated by the attacks. To visualize the results, we chose the smallest ϵ for each of our attacks, under which our model showed at least 80% misclassification. You can see the visualizations in the Figures 2 and 3. We can see that in the examples generated by the FGSM attack, although the original image is still well visible, the perturbation is easily detectable to us. For PGD, AutoPGD, and FW + Dual LMO attacks, however, the perturbations are hardly visible. In fact, from Fig. 3 it is noticeable that PGD and AutoPGD attacks apply small perturbations uniformly over the image. While the FW + Dual LMO attack perturbs only small portions of the image, the perturbations are much more visible.

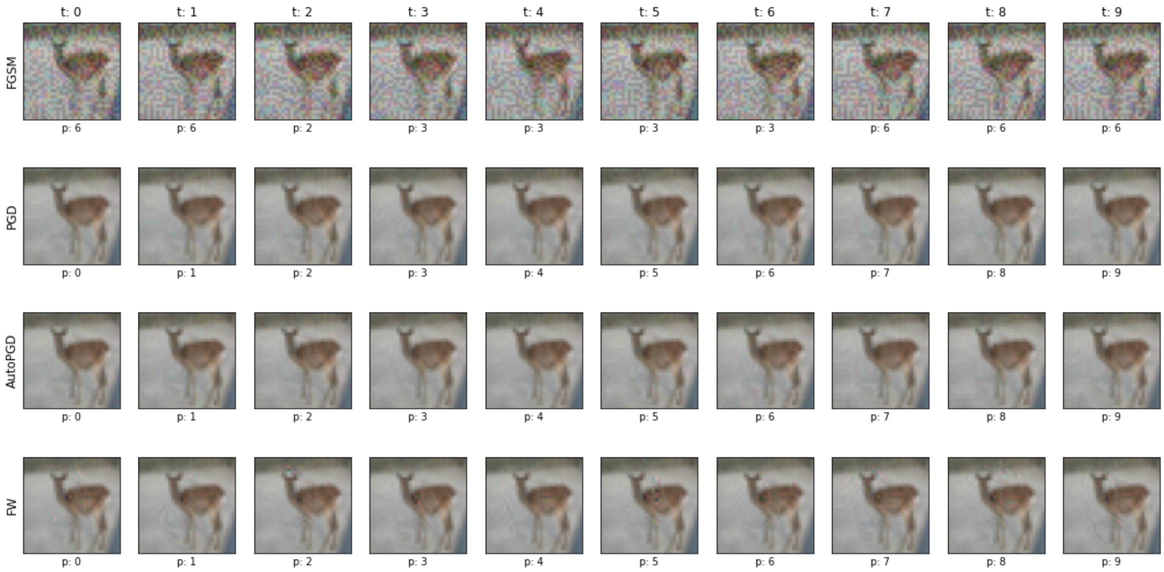


Fig. 2. Adversarial samples on an image with original label 4 (deer).

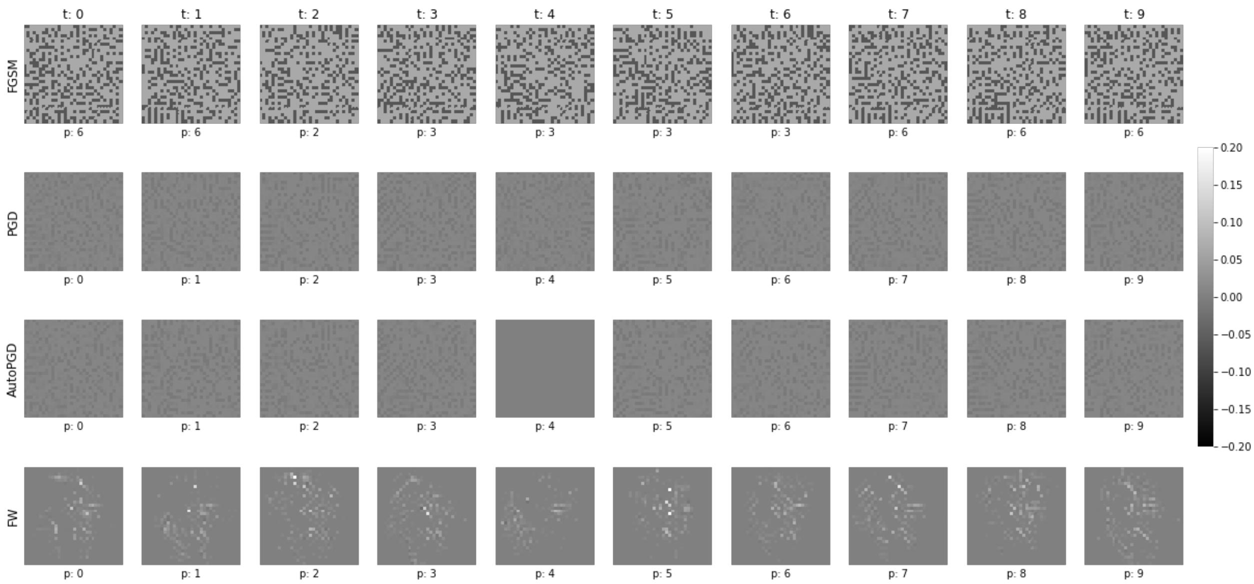


Fig. 3. Perturbations added to the image with original label 4 (deer).

5. Conclusion and Future Work

We compared different attack methods with different metrics. The champion of the comparison is the PGD attack. Although being a very simple attack, it performs very well in terms of misclassification and targeted misclassification rates, is fast, and is almost non-detectable by the human eye in our experiments. AutoPGD, while yielding similar results, is much slower, and hence, comes in second place in our comparison. FW + Dual LMO attack performed very well in terms of duration, misclassification, and targeted misclassification rates, but the perturbations were much more noticeable. The FGSM attack was the fastest with a high misclassification rate but came last in terms of imperceptibility.

Since we've covered only a small portion of attacks, we plan to extend the attack list by adding more well-known or state-of-the-art methods and extend the experiment domain to black-box attacks as well. Furthermore, we plan to test these attacks on a defended model and compare their performances. Particularly, we are interested to see the difference between AutoPGD and PGD attacks on a defended model.

References

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[2] Q. Li, Y. Guo, and H. Chen, "Practical no-box adversarial attacks against dnns," *Pre-proceedings - Advances in Neural Information Processing Systems*, vol. 33, 2020.

[3] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017.

[4] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," *arXiv preprint arXiv:2003.01690*, 2020.

[5] K. Wu, A. Wang, and Y. Yu, "Stronger and faster wasserstein adversarial attacks," in *International Conference on Machine Learning*, pp. 10377–10387, PMLR, 2020.

[6] E. Wong, F. R. Schmidt, and J. Zico Kolter, "Wasserstein adversarial examples via projected sinkhorn iterations," in *36th International Conference on Machine Learning, ICML 2019*, 2019.

[7] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in neural information processing systems*, vol. 26, pp. 2292–2300, 2013.

[8] M. Frank, P. Wolfe, *et al.*, "An algorithm for quadratic programming," *Naval research logistics quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[10] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[11] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. Molloy, and B. Edwards, "Adversarial robustness toolbox v1.2.0," *CoRR*, vol. 1807.01069, 2018.

Սպիտակ տուփով, թիրախավորված մրցակցային հարձակումների համառոտ համեմատությունը խորը նեյրոնային ցանցերում

Գրիգոր Բեզիրգանյան և Հենրիկ Սերգոյան

Մյունխենի Տեխնիկական Համալսարան
e-mail: grigor.bezirganyan@tum.de, henrik.sergoyan@tum.de

Ամփոփում

Այսօր նեյրոնային ցանցերն օգտագործվում են տարբեր ասպարեզներում, որոնցից շատերում կարևոր է ունենալ հուսալի և ճշգրիտ արդյունք: Ահա թե ինչու մրցակցային հարձակումները նեյրոնային ցանցերը դարձնում են ավելի քիչ հուսալի՝ բարձր անվտանգության մակարդակ պահանջող տիրույթներում: Հետևաբար, կարևոր է ուսումնասիրել հարձակման հնարավոր մեթոդները՝ ավելի կայուն և անվտանգ ցանցեր մշակելու համար: Այս հոդվածում մենք քննարկում ենք չորս սպիտակ?տուփով, թիրախավորված մրցակցային հարձակումներ և համեմատում դրանք իրենց սխալ դասակարգման աստիճանի, նպատակային սխալ դասակարգման արագության աստիճանի, հարձակման տևողության և աննկատելիության առումով: Մեր նպատակն է գտնել հարձակում(ներ), որոնք արդյունավետ կլինեն և կստեղծեն մրցակցային օրինակներ՝ փոքր շեղումներով և աննկատելի մարդու աչքի համար:

Բանալի բառեր՝ մրցակցային հարձակումներ, կայունություն, մեքենայական ուսուցում,խորը ուսուցում:

Краткое сравнение между "Белым ящиком", целевыми состязательными атаками противника в глубоких нейронных сетях

Григор Безирганян и Генрик Сергоян

Технический Университет Мюнхена
e-mail: grigor.bezirganyan@tum.de, henrik.sergoyan@tum.de

Аннотация

Сегодня нейронные сети используются в различных областях, в большинстве из которых важно иметь надежный и правильный вывод. Вот поэтому состязательные атаки делают глубокие нейронные сети менее надежными для использования в областях, где безопасность имеет решающее значение. Следовательно, важно изучить потенциальные методы атаки, чтобы иметь возможность разрабатывать гораздо более надежные сети. В этой статье мы рассматриваем четыре "белых ящика" - целенаправленные состязательные атаки и сравниваем их с точки зрения частоты ошибочных классификаций, частоты целевых ошибочных классификаций, длительности атаки и незаметности. Наша цель - найти атаки, которые были бы эффективны и генерировали бы состязательные выборки с небольшими возмущениями и не обнаруживались бы человеческим глазом.

Ключевые слова: состязательные атаки, надежность, машинное обучение, глубокое обучение.

Mathematical Problems of Computer Science 58, 52–60, 2022.
doi: 10.51408/1963-0092

UDC 004.891.3

ininflueDeveloping Aerial Unmanned Effective Decision Makers

Sedrak V. Grigoryan and Edward M. Pogossian

Institute for Informatics and Automation Problems of NAS RA
e-mail: addressforsd@gmail.com, epogossi@aua.am

Abstract

Unmanned aerial vehicles (UAVs, drones) and similar unmanned units are becoming more and more involved in various spheres, such as agriculture, emergency situations, battles, etc. however, in decision making there are still a lot they can be improved to avoid human direct involvement in those problems.

To advance in the problem we develop tools to make UAV autonomously effective decision makers, particularly, able to analyze properly given situations and then according to assigned goals select appropriate strategies to achieve the goals.

In the following work we aim to provide a solution for a single UAV which is able to discover units of interest, and select the target to track, manipulate or hit based on expert specified knowledge, as well as discuss further steps.

Keywords: Object, detection, Decision making, Combinatorial problems, Expert knowledge.

Article info: Received 16 August 2022; accepted 26 September 2022.

1. Introduction

1.1. Problems of Space of UAV Involvements

Involvement of programmatic solutions in various types of UAV-based environments, such as agriculture, emergency situations, battles, and other types of urgent problems, is important and actual problem.

Representation of problems can vary from one to another, while given situation for UAV-based solutions may stay in scope of the following list: maps, emergencies, opponents, their positions, etc.

Overall, it is becoming very important to avoid human involvement in these tasks directly to avoid human causalities, to provide descent support and amount of units involved, thus it is important having decision making modules.

A non-expensive UAV which is able to process the field situation as an image from the top, and make decisions based on the current situation without human involvement is an urgent problem. The advantage of such unit is that it can cost low and has pretty high accuracy and effectiveness.

1.2. Programmatic Improvements of UAV Units

Various tasks can be considered in this space, including:

1.2.1. The tasks of adequate processing of situations. The program has to properly capture and parse the current situation based on retrieved data, mostly from images. This is currently not fully solved, however there are some available solutions for certain types of such tasks, e.g. detecting units of interests, such as emergency areas, e.g. fire sources on the images, etc.

Such solutions require:

a. sufficient preliminary inherited knowledge and ongoing data related to the units on the field to be recognized, particularly the ones to identify the own and opponent units, targeting items, tracking objects, etc.

b. proper training and examining the functionality of target models in performing parsing of situations and recognizing there all valuable units (the mistakes might be very costly depending on the problem).

1.2.2. Making valuable decisions in situations UAV can:

a. analyze them to select with respect to (wrt) the goals the most prospective and simultaneously available ones

b. select plans of attaining those targets

c. analyze compositions of actions, strategies for the perspective plans

d. make evaluation of the strategies and perform appropriate strategies to attain the goals.

1.3. To examine our approach, we concentrate on the topic for a battle field strategy games G, which provide good way to track situation from the top (similar to UAV images).

We consider this as a problem of certain combinatorial RGT class, where the space of solutions is reproducible game trees [1-8].

RGT problems are specified as follows:

• there are (a) interacting actors (players, competitors, etc.) performing (b) identified types of actions in the (c) specified types of situations;

• there are identified utilities, goals for each actor;

• actions for each actor are defined

• the scope of solutions at the situations are fully determined by them (i.e., are identified as games with perfect information)

Actors perform their actions in specified periods of times and do affect situations by actions in time t by transforming them to new situations in time t+1 trying to achieve the best utilities on that situation (goals) by regularities defining these actions.

For example, a way to interpret battle field game G as the RGT problem is:

1. The battling sides can be considered as interacting actors
2. Military units' movements, attacks can be considered as actions
3. The battle field area including military units can be considered as the situations
4. Different situations can be considered as goals: capture objects, destroy enemy units, push frontline.
5. The analysis of given situations are sufficient for selection of proper strategies

1.4. Advances of RGT Solvers

1.4.1. There are certain important advances and achievements in cognizers (RGT Solvers) [7] development:

In it was shown that RGT problems are reducible to each other, particularly, to some standard kernel RGT problem K, say, chess, thus, we get an opportunity to integrate the best-known achievements in solving particular RGT problems into RGT Solvers letting us to apply those achievements to any of RGT problem [1].

In RGT solutions, we follow the research lines of Botvinnik, Pitrat, Wilkins and ones successfully started since 1957 in the Institute for Informatics and Automation Problems at the Academy of Sciences of Armenia and based on modeling of expert approaches involving: knowledge bases, knowledge-based algorithms of decision making and matching situations to classifiers, as well as algorithms of revealing and modifying knowledge.

The advances in RGT [1-10] include the following:

1. Solutions for transforming situations for RGT problems, a solution for chess is available. “Generals: Command and Conquer” game is considered as a sample battlefield problem and positive results were achieved for recognition of military units.
2. Knowledge presentation and matching algorithms were developed generally for RGT problems and adequacy was experimented for chess, marketing and other RGT problems.
3. Planning and decision-making algorithms, IGAF and PPIT (including TZT) based on Botvinnik’s ideas were developed and tested for network intrusion protection problems and chess problems. Additionally, partial implementation of PPIT algorithms were integrated in general RGT Solver and experimented for chess and other RGT problems.

Various urgent combinatorial problems were investigated as RGT problems including network protection from hacker intrusions [1], single ship defense from various types of attacks [6, 7], chess [2, 4], etc.

1.5. We aim to resolve some of above-mentioned tasks by providing programs for UAV, i.e., autonomously effective decision makers, or agents, particularly for type of games G that will allow to process properly situations of G, then according to assigned goals select appropriate strategies for achieving the goals.

In the current work we concentrate on the following problems:

a. From the input images from UAVs detect and classify units of the game G influential for attaining the goals

b. From the input situation including already classified influential units select target to hit.

2. Units’ detection, classification

Classification of influential units is performed via the recorded images. Popular object detection and image classification methods are now widely based on machine learning solutions, particularly deep convolutional neural networks, e.g. in the following an approach for vehicle detection from aerial images is discussed [11].

In the following we also rely on ML solutions to train a model for influential units’ detection and classification.

2.1. Creation of Classification Dataset

Based on analysis of available data, we collected influential unit images and videos. From the collected data images were revealed to describe influential units for training the model. We made

a grouping of some classes of influential for game G units into one class, later to be classified as the same. This allows to have much less classes and possibly higher detection rate. The created dataset mostly consists of aerial photos, because UAVs mostly take pictures that way.
We have selected 8 groups of influential for game G classes, then created dataset consisting of their aerial images.

2.2. Preparation of Detection Model

Once we have the images as discussed in above section, we prepare it for training models. In the case of the game G unit's detection, the model needs both accuracy and speed, but it is more essential to draw accurate detection conclusions. Some of the studies reveal that YOLO provides better detection and speed combination over other models in various problems, providing real time detection ability [11-16]. Based on the available results we use YOLOv5 as a model to be used for our dataset training and detection.
The trained model gave results of accuracy with the values as follows: detection precision about 80%, recall is close to 60-70% and mAp about 60. The summary is in Fig. 1.

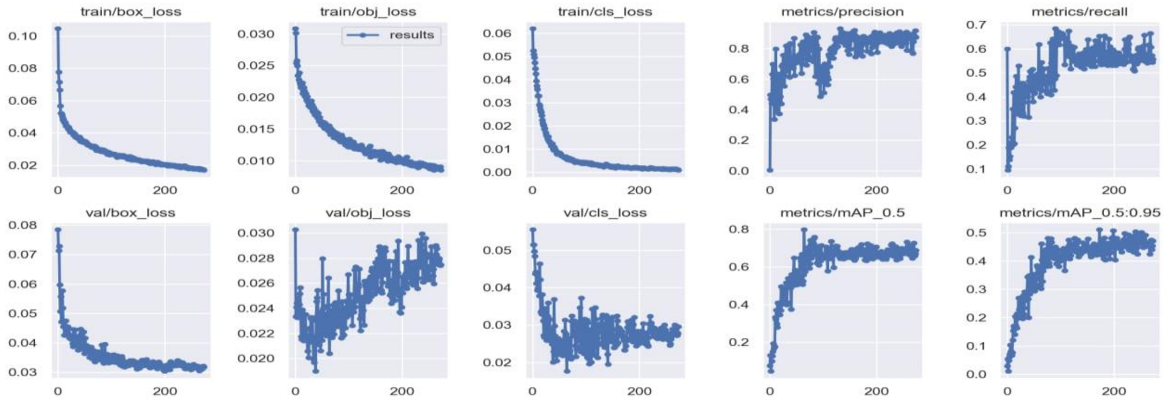


Fig. 1 Metrics of Training Results.

3. Selection of The Target

As described in Introduction chapter we are relying on the achievements of RGT to provide decision making solutions in such problems, particularly the solutions rely on expert knowledge. Here it comes to finding out the knowledge pieces needed in decision making in the game G and specifically for selection a target for UAV managing as we concentrate our attention to that specific game G in the current work.
The experts' analysis and descriptions the following nuclear types [2] of knowledge for game G were revealed.
For the targeting influential units:

1. The class of the G units as classified in section 2, can be reduced to a value in range of {1-8} for each class having a specific value.

2. The price of the unit. This is not the actual cost of the unit, but the price of the unit in the battle, describing how much can its damage be. So we assign this a rule of {1-8} range depending on the type of target.
3. Other types of expert knowledge also participate in decision making, based on which the decision becomes more accurate.

For own UAV:

1. It also contains specific types of knowledge, in this case this is related to the managing abilities, the decision realization instruments type and power, which determine the target to be resolved.

Based on the following nuclear classifiers we construct classes of units that appear as possible

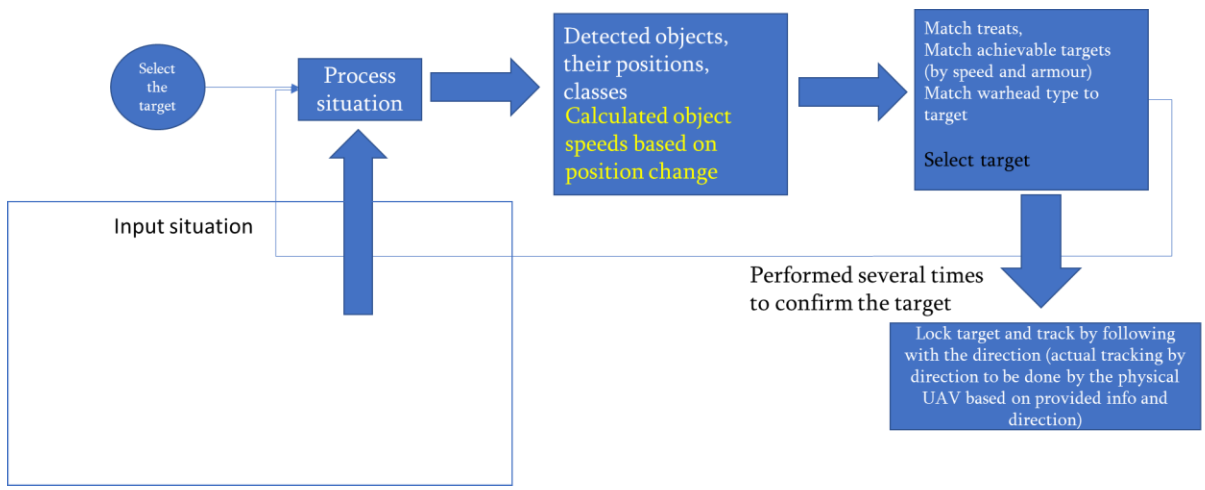


Fig. 2. The Flow of Target Selection Algorithm.

targets [2, 9]. In the tasks we only consider decisions relevant to the game G by the UAV as our own action. So, the simplified version of goal searching algorithms [2, 10] is applied here. First non-perspective targets are filtered out in the situation. Then by unit price the prioritization is applied and, with some additional corrections the target is selected.

Because the situation is changing, the selected target on each situation can be different.

To increase the confidence of correct selection of the targets, in sequential situations the same logic of target selection is applied several times.

If examined target is confirmed, the confidence is increasing. With attaining certain confidence in certain time period, the target is locked on.

3. Above we discussed the basic approach and some of applied knowledge descriptions for the selection of the targets.

The model of detection of influential units and its metrics were provided in section 2.

Knowledge-based approach adequacy has been discussed in [2, 8, 9].

The performance and the efficiency of programs realizing our UAV approach are attributed as follows:

a. The program is developed in python programming language to provide easy and fast transitions between various experimenting environments.
b. To improve efficiency of the program, when the target is selected, it is only tracked without its recurring detection and matching.

c. Once target is locked on, the program calculates and provides the direction for hitting it.
d. The efficiency of the program is experimented with various video inputs with different frame update rate: frames per second (FPS), resolution, the program provides close to real time results: for HD and FullHD videos with 20-25 fps the program is able to achieve close to real time performance.
e. The prepared program and its performance were tested low power-consuming and GPU enhanced devices, which may be a good fit for UAV setup.

4. Future Works

The current solutions demonstrate the positive results of the work, as well as provide background for the future steps.

The next steps of the current works are:

1. The accuracy of the detection of game G units affects the whole flow of target selection and situation processing, decision making, thus improvement of detection is one of essential topics, also due to possible fatal problems in actual application mistakes. For this step we go on the following direction: 1.1. Enhancement of the dataset with new images, 1.2. Enhancement of dataset by machine learning solutions, such as data augmentation, 1.3. Applying machine learning techniques to improve quality of the input 1.4. If the amount of data is sufficient, then classify exact types of units instead of grouping them.
2. Enlarging the scope of considered situation. This assumes enhancing the knowledge for matching situations, which can help in properly selection targets, provide more than one type of actions for involved other than the given single UAV own units, specifying separate targets for own units and the sequence for targets to be hit. The enhancement of knowledge of the experts is an essential part in making decisions and improvement of decision with the increase of expert knowledge is demonstrated in [9], while integration of knowledge-based decision-making algorithms provided in [2, 10] also demonstrated their adequacy.

This provides a good background for using the solutions in real UAVs.

5. Conclusions

In the following work an approach to describe battle field problem is discussed, where a way to formalize the problem is given. The following results were achieved:

1. From open sources many photo and video data were analyzed, and images were revealed to create a dataset of G units. The dataset consists of 8 classes, each of them containing a group of units functionally equal to the ones defined by experts, to achieve an acceptable accuracy in detection.
2. YOLOv5 model was used for training a model to detect the selected classes, and the results of model performance were demonstrated.
3. By close cooperation with experts of that field certain types of knowledge to properly select the target to be hit were revealed.
4. Algorithms to select the target based on input images, classified objects on that and the knowledge of the field are developed.
5. Experiments were conducted for low power computing units and close to real time processing efficiency is achieved.

Relying on the results achieved in this work and achievements described in the field of RGT problems, we plan the next steps of the work as follows:

1. Collect more data from available sources, enhance the existing dataset by machine learning tools. This allows to achieve better detection and classification accuracy, as well as makes it possible later to more detailed classification instead of grouping them.
2. Enlarge the scope of included problems to consider also agricultural, emergency and other urgent applications, to provide certain types of actions based on decisions it makes using algorithms developed for RGT Solvers [2, 9, 10].
3. Enhance knowledge base for the problems based on expert knowledge to enable various types of actions, including ways of more appropriate target selections, target managing sequence selections, etc.

References

[1] E. Pogossian, A. Javadyan and E. Ivanyan., "Effective discovery of intrusion protection strategies" *The International Workshop on Agents and Data Mining, Lecture Notes in Computer Science*, St. Petersburg, Russia, pp. 263-274, 2005.

[2] S. Grigoryan, *Research and Development of Algorithms and Programs of Knowledge Acquisition and Their Effective Application to Resistance Problems*, PhD, Yerevan, Armenia, 2016.

[3] S. Grigoryan, “On validity of personalized planning and integrated testing algorithms in reproducible games”, *Proceedings of International Conference in Computer Sciences and Information Technologies*, Yerevan, Armenia, pp. 317-321, 2015.

[4] N. Hakobyan, “A system for transforming images to symbolic presentation for combinatorial defense and competition problems”, *ISSN 0002-306X. Proc. of the RA NAS and NPUA Ser. of tech. sc.,* . vol. 72, no. 2, pp. 199-209, 2019.

[5] E. Pogossian, “Effectiveness enhancing knowledge-based strategies for SSRGT class of defense problems”, *NATO ASI 2011 Prediction and Recognition of Piracy Efforts Using Collaborative Human-Centric Information Systems*, Salamanca, Spain, 2011.

[6] D. Dionne, E. Pogossian, A. Grigoryan, J. Couture and E. Shahbazian, “An optimal sequential optimization approach in application to dynamic weapon allocation in naval warfare”, *11th International Fusion 2008 Conference in Cologne*, July 1-3, 2008.

[7] E. Pogossian, D. Dionne, A. Grigoryan, J. Couture and E. Shahbazian, “Developing goals directed search models empowering strategies against single ownship air threats”, *CSIT2009: International Conference in Computer Sciences and Information Technologies*, Yerevan, Armenia, 5 p. 2009.

[8] E. Pogossian, “Towards adequate constructive models of mental systems”, *CSIT2017: International Conference in Computer Science and Information Technologies*, Yerevan, Armenia, pp. 96-101, 2017.

[9] E. Pogossian, *Constructing Models of Being by Cognizing*, Academy of Sciences of Yerevan, Armenia, 2020.

[10] M. Buniatyan, *Developing Software for Expert Knowledge/Classifiers-Based Effective Strategies Formation and Applications*, Master thesis, IIAP NAS RA, Yerevan, Armenia, 2022.

[11] F. Li, S. Li, C. Zhu, X. Lan and H. Chang, “Cost-effective class-imbalance aware CNN for vehicle localization and categorization in high resolution aerial images”, *Remote Sensing*, vol 9, no. 5, pp. 1-29, 2017.

[12] K. Zhang, Ch. Wang, X. Yu, et al. “Research on mine vehicle tracking and detection technology based on YOLOv5”, *Systems Science and Control Engineering*, vol. 10, no. 1, pp. 347-366, 2022.

[13] YOLOv5 repository [Online] Available: https://github.com/ultralytics/yolov5

[14] N. Sabina, M. Aneesa and P. Haseena, “Object detection using YOLO and mobilenet SSD: A comparative study”, I*nternetional Journal of Engineering Research&Technology*, vol. 11, no. 6, 134-138, 2022.

[15] K. R. Ahmed and P. Smart, “Detection using deep learning based on dilated convolution”, Sensors, 21, 24, 8406, 2021. Doi: 10.3390/s21248406

[16] A. Bochkovskiy, Ch. Wang and M. Liao, “YOLOv4: Optimal speed and accuracy of object detection”. arXiv:2004.10934v1, 2020.

Անօդաչու արդյունավետ որոշումների կայացման ծրագրերի մշակում

Սեդրակ Վ. Գրիգորյան և էդվարդ Մ. Պողոսյան

ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ
e-mail: addressforsd@gmail.com, epogossi@aua.am

Ամփոփում

Անօդաչու թռչող սարքերը (ԱԹՍ, դրոն), եւ այլ անօդաչու միավորները լայն կիրառում են ստանում են տարատեսակ կարեւոր ոլորտներում, ինչպիսիք են՝ գյուղատնտեսությունը, արտակարգ իրավիճակները, ռազմական խնդիրները եւ այլն, չնայած դրանց որոշումների կայացման եղանակներում դեռ կան լավարկման հնարավորություններ՝ խուսափելու համար մարդկային գործոնի ուղղակի ներգրավվածությունից:

Այսպիսի խնդիրներում առաջադիմելու նպատակով մենք մշակում ենք գործիքներ, որոնք հնարավորություն կտան ԱԹՍների կողմից ինքնուրույն արդյունավետ որոշումներ կայացնել, մասնավորապես՝ վերլուծելով ստեղծված իրավիճակը, ըստ հասցեագրված նպատակների մշակել ռազմավարություն՝ այդ նպատակներին հասնելու համար:

Այս աշխատանքում մենք ձգտում ենք տալ մի լուծում միայնակ ԱԹՍի համար լուծում, որը կհայտնաբերի իրավիճակում հետաքրքրություն ներկայացնող միավորները, դրանցից կընտրի թիրախ հետեւելու, խոցելու կամ այլ նպատակի համար՝ հիմնված փորձագիտական գիտելիքների վրա: Հաջորդիվ նաեւ բերվում են լուծման հետագա զարգացման քայլերը:

Բանալի բառեր՝ օբյեկտների հայտնաբերում, որոշումների կայացում, կոմբինատոր խնդիրներ, փորձագիտական գիտելիքներ:

Разработка программ принятия эффективных беспилотных решений

Седрак В. Григорян и Эдвард М. Погосян

Институт проблем информатики и автоматизации НАН РА
e-mail: addressforsd@gmail.com, epogossi@aua.am

Аннотация

Беспилотные летательные аппараты (БПЛА, дроны) и подобные беспилотные устройства наряду с возрастающим числом приложений в сельском хозяйстве, управлении при чрезвычайных ситуациях, например боевых и т.д., требуют значительного усовершенствования эффективного принятия решений.

Нами разрабатывается програмы, позволяющие, в частности, анализировать ситуации, а затем в соответствии с поставленными целями выбирать подходящие стратегии для достижения целей.

В работе представлены описание процедуры анализа ситуации для обнаружения целевых обьектов и их отслеживания, анализа версий решений с использованием наличных знаний эксперта, выбора конкретной цели и принятия окончательного решения.

Ключевые слова: обнаружение объектов, принятие решений, комбинаторные задачи, экспертные знания.

Mathematical Problems of Computer Science 58, 61–66, 2022.
doi: 10.51408/1963-0093

UDC 510.64

Proof Complexity of Hard-Determinable Balanced Tautologies in Frege Systems

Anahit A. Chubaryan

Yerevan State University
e-mail: achubaryan@ysu.am

Abstract

Hard-determinable property and balanced property of tautologies are specified as important properties in the study of proof complexities formerly. In this paper hard-determinable and balanced properties are studied together. It is shown that some sequences of hard determinable balanced tautologies have polynomially bounded Frege proofs.

Keywords: Hard-determinable tautologies, Balanced tautologies, Frege systems, Proof complexity characteristics.
Article info: Received 29 June 2022; accepted 29 September 2022.

1. Introduction

One of the most fundamental problems in proof complexity theory is to find an efficient proof system for classical propositional logic (CPL). There is a widespread understanding that polynomial time computability is the correct mathematical model of feasible computation. According to the opinion, a truly "effective" system should have a polynomial - size $p(n)$ proof for every tautology of size n. In [1] Cook and Reckhow named such a system a *supersystem*. They showed that $NP = coNP$ iff there exists a supersystem. It is well known that many systems are not super. This question about the Frege system, the most natural calculi for propositional logic, is still open. In many papers, some specific sets of tautologies are introduced, and it is shown that the question about polynomial bounded sizes for Frege proofs of all tautologies is reduced to an analogous question for a set of specific tautologies. In particular the *hard-determinable* tautologies and *balanced* tautologies are introduced in [2,3] as such sets of specific tautologies. In this paper, the hard-determinable and balanced properties are studied together and it is shown that some sequences of hard-determinable balanced tautologies have polynomial bounded Frege proofs. Using the notions and results of this paper and the results of [3-4] the above-mentioned statement of Cook and Reckhow can be rephrased as follows: $NP = coNP$ iff in some Frege system of CPL the proofs for all hard-determinable balanced formulas are polynomially bounded.

2. Preliminaries

To prove our main result, we recall some notions and notation. We will use the current concepts of the unit Boolean cube (E^n), a propositional formula, a tautology, a proof system for CPL, and proof complexity. The particular choice of a language for presenting propositional formulas is immaterial in this consideration. However, because of some technical reasons we assume that the language contains propositional variables, denoted by small Latin letters with indices. Logical connectives ¬, &, ∨, ⊃, and parentheses (,). Note that some parentheses can be omitted in generally accepted cases.

2.1. Hard-determinable and Balanced Tautologies

Following the usual terminology we call the variables and negated variables *literals*.

The conjunct K (clause) can be represented simply as a set of literals (no conjunct contains a variable and its negation simultaneously).

In [3] the following notion is introduced.

We call each of the following trivial identities for a propositional formula ψ a *replacement-rule*:

$$0\&\psi = 0,\quad \psi\&0 = 0,\quad 1\&\psi = \psi,\quad \psi\&1 = \psi,\quad \psi\&\psi = \psi,\ \psi\&\neg\psi = 0,\ \neg\psi\&\psi = 0,$$
$$0\vee \psi = \psi,\ \psi\vee 0 = \psi,\ 1\vee\psi = 1,\ \psi\vee 1 = 1,\ \psi\vee\psi = \psi,\ \psi\vee\neg \psi = 1,\ \neg\psi\vee\psi = 1,$$
$$0\supset\psi = 1,\ \psi\supset 0 = \neg\psi,\ 1\supset\psi = \psi,\ \psi\supset 1 = 1,\ \psi\supset\psi = 1,\ \psi\supset\neg\psi = \neg\psi,\ \neg\psi\supset \psi = \psi,$$
$$\neg 0 = 1, \neg 1 = 0, \neg\neg\psi = \psi.$$

Application of a replacement rule to certain word consists in replacing some its subwords, having the form of the left-hand side of one of the above identities by the corresponding right-hand side. Let φ be a propositional formula, let $P = \{p_1, p_2, \dots, p_n\}$ be the set of the variables of φ, and let $P' = \{p_{i_1}, p_{i_2}, \dots, p_{i_m}\}$ $(1 \le m \le n)$ be some subset of P.

Definition 1. *Given $\sigma = \{\sigma_1, \sigma_2, \dots, \sigma_m\} \in E^m$, the conjunct $K^\sigma = \{p_{i_1}^{\sigma_1}, p_{i_2}^{\sigma_2}, \dots, p_{i_m}^{\sigma_m}\}$ is called φ-determinative if assigning σ_1 $(1 \le j \le m)$ to each p_{ij} and successively using replacement rules we obtain the value of φ (0 or 1) independently of the values of the remaining variables.*

Definition 2. *We call the minimal possible number of variables in a φ-determinative conjunct the* ***determinative size*** *of φ and denote it by $ds(\varphi)$.*

By $|\varphi|$ we denote the size of the formula φ, defined as the number of all logical signs entries in it. It is obvious that the full size of the formula, which is understood to be the number of all symbols is bounded by some linear function in $|\varphi|$.

Definition 3. *For sufficiently large n the tautologies φ_n are called hard-determinable if there is some constant c such that $log_{|\varphi_n|} ds(\varphi_n) \to c$ for $n \to \infty$.*

Definition 4. *A formula φ is* ***balanced*** *if every propositional variable occurring in φ occurs exactly twice, once positive and once negative.*

Example 1. The tautologies $\varphi_n = p_1 \supset (p_1 \supset (p_2 \supset (\neg p_2 \supset (\ldots \supset (p_n \supset p_n) \ldots))))$ are balanced. It is not difficult to see that $ds(\varphi_n) = 1$, hence φ_n are not hard-determinable.

Example 2. The tautologies $QHQ_n = V_{0\le i\le n} \&_{1\le j\le n} \left[V_{1\le k\le i} \bar{q}_{i,j,k} \vee V_{i<k\le n} q_{k,j,i+1}\right] (n \ge 1)$, are balanced. Put $Q_{i,j} = V_{1\le k\le i} \bar{q}_{i,j,k} \vee V_{i<k\le n} q_{k,j,i+1} (n \ge 1,\ 0 \le i \le n,\ 1 \le j \le n)$, then $QHQ_n = V_{0\le i\le n}(Q_{i1} \& Q_{i2} \& \ldots \& Q_{ij} \& \ldots \& Q_{i(n-1)} \& Q_{in})$ and therefore $ds(QHQ_n)$. It is not difficult to see, that $|QHQ_n| = \frac{3n^2(n+1)}{2} - 1$ |, hence QHQ_n are hard-determinable as well.

2.2. Proof Systems and Proof Complexities

Let us recall some notions from [1].

A Frege system $\boldsymbol{\mathcal{F}}$ uses a denumerable set of propositional variables, a finite, complete set of propositional connectives; $\boldsymbol{\mathcal{F}}$ has a finite set of inference rules defined by a figure of the form $\frac{A_1 A_2 \ldots A_m}{B}$ (the rules of inference with zero hypotheses are the schemes of axioms); $\boldsymbol{\mathcal{F}}$ must be sound and complete, i.e. for each rule of inference $\frac{A_1 A_2 \ldots A_m}{B}$ every truth-value assignment, satisfying $A_1 A_2 \ldots A_m$, also satisfies B, and $\boldsymbol{\mathcal{F}}$ must prove every tautology.

In the theory of proof complexity two main characteristics of the proof are: l – complexity to be the size of a proof (= the sum of all formulae sizes) and t – complexity to be its length (= the total number of lines). The minimal l – complexity (t – complexity) of a formula φ in a proof system Φ we denote by $l_\varphi^\Phi (t_\varphi^\Phi)$.

The ***polynomial equivalence*** ($p - l$ --equivalence, $p - t$ --equivalence) of two proof systems by some proof complexity measure means that the transformation of any proof in one system into a proof in another system can be performed with no more than polynomial increase of proof complexity measure.

It is well known that any two Frege systems are $p - l$ -equivalent ($p - t$ -equivalent).

Let M be some set of tautologies.

Definition 5. *We call the Φ-proofs of tautologies from the set M t -polynomially (l – polynomially) bounded if there is a polynomial $p()$ such that $t_\varphi^\Phi \le p(|\varphi|)(l_\varphi^\Phi \le p(|\varphi|))$ for all φ from M.*

2.3. Former Results

It was previously proven that

a) tautologies without hard-determinability condition have t -polynomially (l - polynomially) bounded proofs in all systems of CPL [4],
b) hard-determinability condition is sufficient (but not necessary) to obtain exponential lower bounds for both proof complexities of tautologies in "weak" proof systems of CPL (Cut-free sequent, Resolution, Cutting planes etc.) [4],
c) hard-determinability condition is not sufficient for exponential lower bounds of proof complexities in Frege systems: for some examples of hard-determinable formulas the t - polynomially (l - polynomially) bounded Frege-proofs are given in [2].

Some proof systems of CPL (calculus of structures with deep inference rules), where the author considers only formulas in negation normal form, are studied in [3], where among the rest of the results it is proved that

a) the set of above mentioned balanced formulas QHQ_n have polynomially bounded proofs in one of the studied system sKS,
b) the relations between the proof complexities in the system sKS and the Frege systems are unknown for the present.

3. Main Result

Let F be some Frege system with inference rule *modus ponens*.

Theorem 1. *The F -proofs of tautologies QHQ_n $(n \geq 1)$ are t-polynomially (t-polynomially) bounded.*

To prove, we use the method of [2] for description of some polynomially bounded proof of QHQ_n direct in F by reducing it to F -proofs of well-known tautologies

$$PHP_n = \&_{0\leq i\leq n} V_{1\leq j\leq n} p_{ij} \supset V_{0\leq i<k\leq n} V_{1\leq j\leq n} \left(p_{ij} \& p_{kj}\right) (n \geq 1)$$

presenting the Pigeonhole Principle . It is proved in [5] that the set of these formulas is t-polynomially (l- polynomially) bounded.

The following two auxiliary statements will be of use:

Lemma 1. *Given arbitrary formulas α, β, γ, α_i, β_i, α_{ij} and β_{ij}, the F-proofs of the following tautologies are t-polynomially (l-polynomially) bounded:*

1) $\alpha \vee \alpha^{-}$,
2) $(\alpha \supset \beta) \supset ((\beta \supset \gamma) \supset (\alpha \supset \gamma))$,
3) $(\beta^{-} \supset \alpha) \supset ({}^{-}\alpha \supset \beta)$,
4) $\alpha_1 \supset (\alpha_2 \supset (\ldots \supset (\alpha_k \supset \alpha_1 \& \alpha_2 \& \cdots \& \alpha_k)\ldots))$ $(k \geq 2)$,
5) $\alpha \vee \alpha^{-} \supset \beta_1 \vee \cdots \vee \beta k \vee \alpha \vee \beta k+1 \vee \cdots \vee \beta k+r \vee \alpha^{-} \vee \beta k+r+1 \vee \cdots \vee \beta k+r+t$ $(k \geq 1, r \geq 1, t \geq 1)$,
6) $\neg(V_{1\leq i\leq k} \&_{1\leq j\leq m} \alpha_{ij}) \supset \&_{1\leq i\leq k} V_{1\leq j\leq m} \bar{\alpha}_{ij}$ $(k \geq 1, m \geq 1)$
7) $\&_{1\leq i\leq k}(\beta_{1i} \vee \beta_{2i}) \supset \neg(V_{1\leq i\leq k}(\bar{\beta}_{1i} \& \bar{\beta}_{2i}))$ $(k \geq 1)$.

The proof is obvious.

Lemma 2. *Let Q_{ij} and Q_{kj} $(0 \leq i<k \leq n, 1 \leq j \leq n)$ be the above denoted subformulas of QHQ_n, then F-proofs of the formulas $Q_{ij} \vee Q_{kj}$ be t-polynomially (l-polynomially) bounded.*

The proof follows from the fact of existence of some s and m $(1 \leq s \leq n, 1 \leq m \leq n)$ such that Q_{ij} contains q_{sjm} and Q_{kj} contains $\neg q_{sjm}$, and also from 1) and 5) of Lemma 1.
From 6) of Lemma 1 we infer for the formula $Q_n = V_{0\leq i\leq n} \&_{1\leq j\leq n} Q_{ij}$.

Condition 1. *The F-proofs of the formulas*

$$\neg QHQ_n \supset \&_{0\leq i\leq n} V_{1\leq j\leq n} \neg Q_{ij}$$

are t-polynomially (l-polynomially) bounded.
Put

$$PHP_n' = \&_{0\leq i\leq n} V_{1\leq j\leq n} \neg Q_{ij} \supset V_{0\leq i<k\leq n} V_{1\leq j\leq n} \neg\left(Q_{ij} \& \neg Q_{kj}\right) \quad (1)$$

The formulas (1) are obtained from the PHP_n by the corresponding substitutions. Hence,

Condition 2. *The F-proofs of the formulas (1) are t-polynomially (l-polynomially) bounded.*
Let

$$A_n = V_{0\le i<k\le n} V_{1\le j\le n}(\neg Q_{ij} \& \neg Q_{kj}).$$

Using conditions (1), (2), and item 2) of Lemma 1, we obtain

Condition 3. *The F-proofs of the formulas $\neg\ QHQ_n \supset A_n$ are t-polynomially (l-polynomially) bounded.*

From Lemma 2 and item 4) of Lemma 1 we have

Condition 4. *The F-proofs of the formulas*

$$B_n = \&_{0\le i<k\le n} \&_{1\le j\le n}(Q_{ij} \vee Q_{kj})$$

are t-polynomially (l-polynomially) bounded, and from item 7) of Lemma 1 it follows that the F-proofs of the formulas $\neg A_{n,m}$ are t-polynomially (l-polynomially) bounded as well.

From the conditions (3), (4), and item 3) of Lemma 1 we have a *t*-polynomial (*l*-polynomial) bound for the *F*-proofs of Q_n .

Corollary1. There are hard-determinable balanced formulas the *F*-proofs of which are *t*-polynomially (*l*-polynomially) bounded.

4. Conclusion

Using the polynomial equivalence of different Frege systems [1], the above mentioned result of Cook and Reckhow can be rephrased as follows: $NP = coNP$ iff in some Frege system of CPL the proofs for all hard-determinable balanced formulas are polynomially bounded.

References

[1] S. A. Cook and A. R. Reckhow, “The relative efficiency of propositional proof systems,” *J. Symbolic Logic*, vol. 44, pp. 36–50, 1979.
[2] S. R. Aleksanyan and A. A. Chubaryan, “The polynomial bounds of proof complexity in Frege systems”, *Siberian Mathematical Journal*, Springer Verlag, vol. 50, no. 2, pp. 243-249, 2009.
[3] L. Sraßburger, “Extension without cut”, *Annals of Pure and Applied Logic*, vol.163, pp. 1995- 2007, 2012.
[4] A. A. Chubaryan, “Relative efficiency of a proof system in classical propositional logic,” *Izv. NAN Armenii Mat*., vol. 37, no. 5, pp. 71–84, 2002.
[5] S. R. Buss, “Polynomial size proofs of the propositional pigeonhole principle,” Journal Symbolic Logic, vol. 52, pp. 916–927, 1987.

Դժվար-որոշելի բալանսավորված նույնաբանությունների արտածումների բարդությունները Ֆրեգեի համակարգերում

Անահիտ Ա. Չուբարյան

Երևանի պետական համալսարան

e-mail: achubaryan@ysu.am

Ամփոփում

Նախկինում նույնաբանությունների դժվար-որոշելիության հատկությունը և բալանսավորված լինելու հատկությունը առանձնացվել էին որպես կարևոր հատկություններ արտածումների բարդությունների ուսումնասիրություններում: Այս հոդվածում դժվար-որոշելիության և բալանսավորված լինելու հատկությունները ուսումնասիրվում են համատեղ: Ապացուցվել է, որ դժվար-որոշելի բալանսավորված նույնաբանությունների մեկ դասի համար արտածումները Ֆրեգեի համակարգերում բազմանդամորեն սահմանափակ են:

Բանալի բառեր՝ դժվար-որոշելի նույնաբանություններ, բալանսավորված նույնաբանություններ, Ֆրեգեի համակարգեր, արտածման բարդությունների բնութագրիչներ:

Сложности выводов трудно-определяемых балансированных формул в системах Фреге

Аанаит А. Чубарян

Ереванский государственный университет

e-mail: achubaryan@ysu.am

Аннотация

Ранее свойство трудно-определяемости и свойство балансированности тавтологий были выдлены как важные свойства в исследованиях сложностей выводов. В настоящей статье свойства трудно-определяемости и балансированности изучаются совместно. Доказана полиномиальная ограниченность выводов в системах Фреге для некоторого класса трудно-определяемых балансированных формул.

Ключевые слова: трудно-определяемые тавтологии, балансированные тавтологии, системы Фреге, характеристики сложностей выводов.

Mathematical Problems of Computer Science 58, 67–83, 2022.
doi: 10.51408/1963-0094

UDC 004.725, 004.852

Research of Obfuscated Malware with a Capsule Neural Network

Timur V. Jamgharyan

National Polytechnic University of Armenia
e-mail: t.jamgharyan@yandex.ru

Abstract

The paper presents the results of a research of using transfer training of the capsule neural network to detect malware. The research was carried out on the basis of the source code of malware using the context-triggered piecewise hashing method. The source codes of malware were obtained from public sources of software. Verification of the capsule neural network learning results was carried out using a trained convolutional neural network, and publicly available sources of test to malware. The research was conducted on six types of malware. Software source code, part of capsule neural network training datasets, pre-trained capsule neural network, and full research are publicly available at https://github.com/T-JN

Keywords: Capsule neural network, Context triggered piecewise hashing, Edit distance, Intrusion detection system, Transfer learning.

Article info: Received 9 June 2022; accepted 24 November 2022.

1. Introduction

Malware injected into Infrastructure through zero-day vulnerabilities in network equipment is a huge cybersecurity problem. The network infrastructure (NI) protection architecture implies the construction of a multi-level, complementary security system. Part of the NI security design is an intrusion detection system (IDS).

In the studies [1]-[5], the types of IDS, the ways of their application and the mechanisms of their work are considered in detail. «Classic» IDS can be classified as:

- host-based IDS, that is detection of attacks on a specific network node,
- network-based IDS, that is, detecting attacks on the network or its segment.

Existing IDS that do not use machine learning (ML) in their functionality (both proprietary and open source) [6]-[9], have one common drawback: they all respond to the threat that is embedded in the rule sets. There is also a high probability of various false positives: (true positive, true negative, false positive, false negative) [10]. Malware is the most common threat

vector in most operating environments [11]. The IDS software ecosystem offers many utilities and application suites that can help collect signals from all types of network traffic [12].
For IDS operating without the use of ML at different levels of the Open System Interconnection (OSI) model [13], the task of detecting malware modifications was secondary. Basically, the task of detecting and neutralizing malware was assigned to antivirus software. But with the convergence of attacks at different levels of the OSI model and the emergence of software-defined networks (SDN), new types of threats and possible attacks arise, the neutralization of which by «standard» methods is difficult [14]-[15]. New systematic approaches are required to solve these problems. With the increase in the growth of attacks built on the basis of ML and machine-to-machine (M2M), new threats to the NI also arise. The requirements for security systems are increasing. The convergence of system, network and cloud services increases both the «attack surface» [16] and the «attack space power» [17]. Of particular danger are attacks «designed» using ML [18]-[20]. Researchers are working on the application of ML to create and build a new type of IDS [21-25]. Unlike «classic» IDS, built on the basis of ML can be further trained, being in one way or another a malware generator [26]-[28]. At this stage, both conceptually new solutions in the field of ML application in IDS are being developed, as well as improvements to existing ones. The papers [29]-[32] consider the issues of using ML to create one or another type of IDS. Researchers and developers of ML-based IDS are faced with a large number of tasks that need to be solved, due to the novelty of this area of information security.

- The task of having annotated data for training a neural network (Annotation is the process of labeling raw data so that it can become training for machine learning [11]). No algorithm can handle really bad data. There are many different requirements for training datasets, in particular, representativeness and «noiselessness». [33]. Unlike neural networks that process images, sound, text, etc., for which there are verified datasets [34]-[39], datasets for training an IDS must to some extent, consist of malware. Researchers have access to certain resources that supply research malware [40]-[46], but these resources make them public with a delay.
- The task of increasing the learning rate of IDS built on the basis of ML. Unlike other neural networks where the main attention is paid to the quantity and quality of training data, in intrusion detection systems built on the basis of ML, in many cases, the speed of learning is also important. As shown in [47], since the emerging malware not included in any database has a different data distribution compared to the original training samples, the efficiency of model detection will decrease when it encounters new malware.
- The task of correctly calculating the degree of threat in an attack using ML [48]. When developing an IDS based on ML, it is necessary to correctly calculate the degree of threat to the protected NI.

In addition to general tasks, there are also specific tasks: since each group and type of malware requires its own specific detection methods [49]-[50].

- Detection based on signature analysis, where a database of malware hashes is used as a signature,
- Detection based on Indicator of Compromise (IoC). It is a set of artifacts based on which malware can be detected: registry branches, loadable libraries, IP addresses, byte sequences, software versions, date and time triggers, ports involved [51].
- Research based on context triggered piecewise hashing (CTPH), (context triggered piecewise hashing is a method of calculating piecewise hashes from input data [52]). Malware developers use various techniques to change the original malware signature to make hashes harder to detect: encryption, obfuscation, reordering of files and libraries, re-distribution and code building in order to fool the detection system, giving malware a

new look and changing the hash values. In this case, malware remains undetected for some time [53].

Various researchers are considering the use of CTPH techniques for malware detection. In [54], the issue of applying transfer learning to solve the problem of malware domain bias is considered, and in [55], the issue of automatic malware family identification and classification through online clustering is considered. But the main issues of preparing malware datasets and training IDS based on ML remain open.The issue of increasing the performance of an IDS based on ML with a small set of training datasets remains relevant. In this paper, a method for applying transfer learning of a capsule neural network with the calculation of CTPH and editing distance to increase the learning rate and detection of malware is investigated. The Levenshtein method [56] (Equation 1) and the method using the *ssdeep* program [57] were chosen as the mathematical apparatus for calculating the editorial distance. To assess the quality of binary learning, the Matthews correlation (Equation 2) [58] was used. The source codes of the malware for creating a set of annotated datasets were taken from open sources. The following malware was used: *mimikatz, athena, engrat, grum, surtr, dyre.*

$$D(i,j) = \begin{cases} 0, & i = 0,\ j = 0 \\ i, & j = 0,\ i > 0 \\ j, & i = 0,\ j > 0 \\ \min\{ & \\ \quad D(i,j-1)+1,\ j>0,\ i>0 & \\ \quad D(i-1,j)+1+m(M[i],N[j]), & \\ \} & \end{cases} \tag{1}$$

Levenshtein editorial distance calculation equation,

where, D - the editorial distance, M, N- the length of strings obtained as a result of CTPH over some alphabet (in this case HEX), i - remove step from the first line, j-insert into the first line.

$$\phi = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}, \tag{2}$$

where,
ϕ - Matthews correlation
TP - true positive,
TN -true negative,
FP -false positive,
FN - false negative.

A capsule neural network was chosen as a transfer learning model. The choice of the capsule network is due to the following reasons:

- the capsule network does not require a large amount of training data, which is critical for this research,
- the capsule network explores hierarchical relationships, which allows detecting possibly probable versions, in the presence of a primary code (a fragment of the main code) of malware,
- the capsule network allows searching even in obfuscated source code with a minimum malware representativeness value,

- the capsule network is the most easily adaptable to changing the learning algorithm compared to other neural networks.

2. Diagrams of Neural Networks

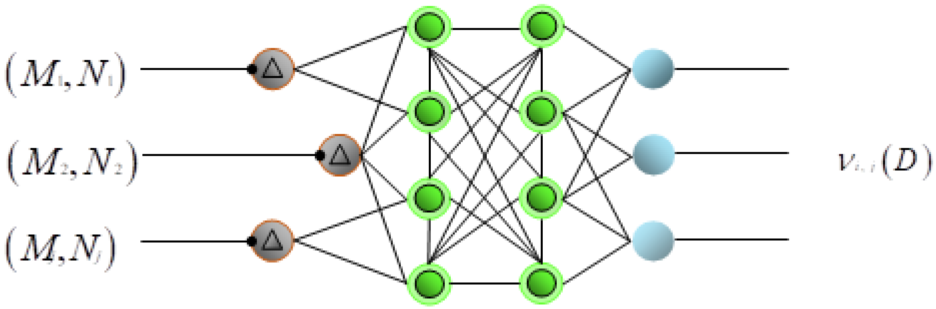
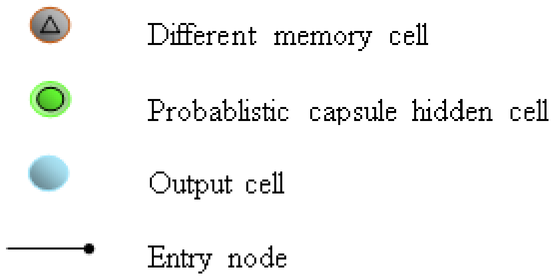


Fig.1. Diagram of a capsule neural network.



The nonlinearity function of the capsule network is determined by (Equation 3) [59].

$$\boldsymbol{v_i} = \frac{||s_i||^2}{1 + ||s_i||^2} \frac{s_i}{||s_i||}, \tag{3}$$

where, s_j- the result obtained in the previous step, $\boldsymbol{v_i}$ - the result obtained after applying the non-linearity. The left side of the equation performs additional compression, and the right side of the equation performs unity scaling of the output vector.

The trained convolutional neural network (Fig. 2) was chosen as a test to check the reliability of the output data. As «weight coefficients» of the convolutional neural network, the value of CTPH was calculated the used *ssdeep* software.

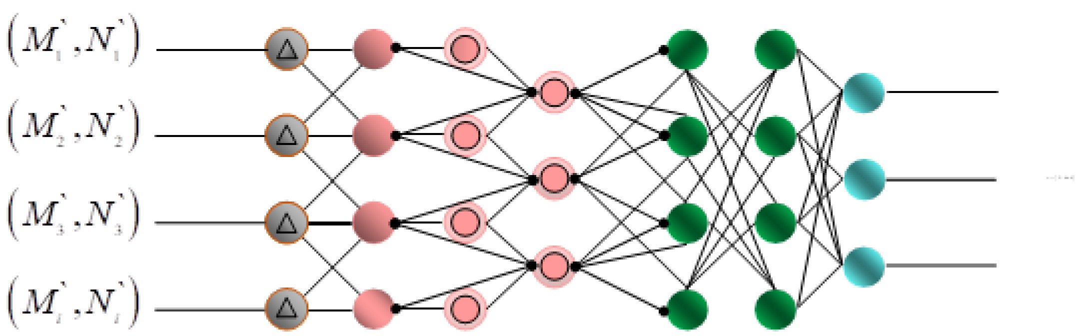
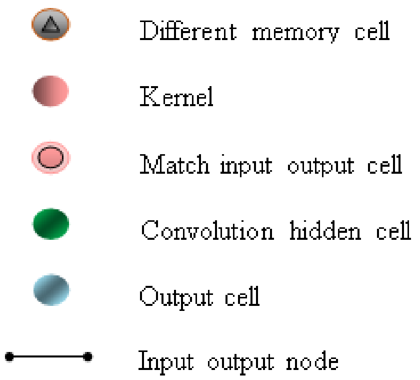


Fig. 2. Diagram of a convolutional neural network.



Verification of the results obtained from both neural networks was carried out using public malware detection services [60]-[61]. The developed software algorithm is shown in Fig.3.

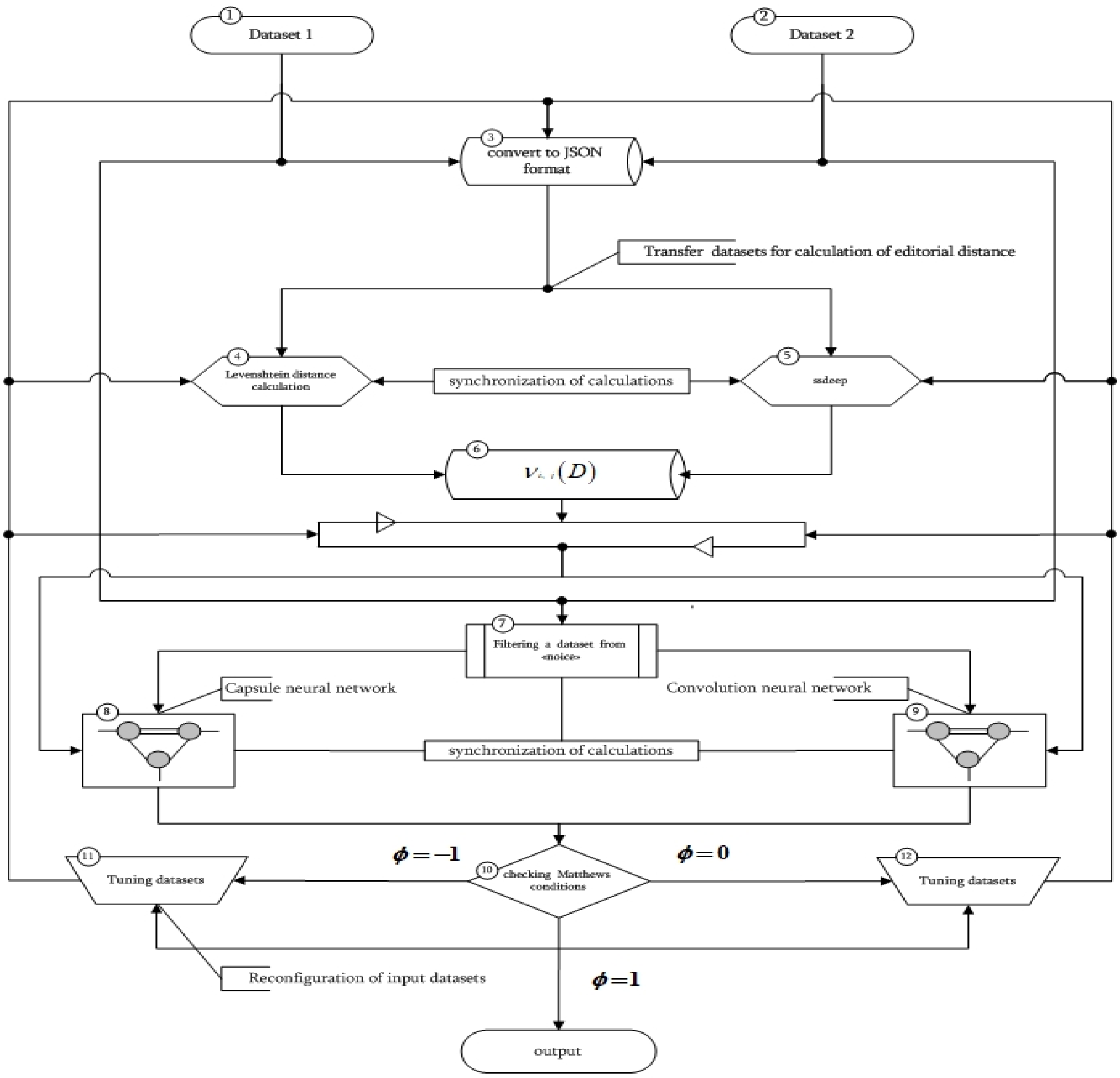


Fig. 3. Algorithm of the developed software.

Algorithm operation:

Operations on the input data of the research.

- The dataset generated from the malware source code was obfuscated using various tools [62]-[63] and prepared for training a capsule neural network (dataset 1).
- The same non-obfuscated dataset (dataset 2) generated from the malware source code was prepared to train a convolutional neural network.

A total of 1000 annotated datasets of various sizes (20.40, 80, 128, 256, 512, 1024 bytes) were prepared for *mimikatz, athena, engrat, grum, surtr, dyre* software.

Steps 1, 2: input of the initial malware dataset into the trained neural networks and the conversion module,

Step 3: converting the source dataset to javascript object notation (JSON) format and setting the CTPH step size,

Step 4: calculation of the edit distance by the Levenshtein method,

Step 5: computation CTPH using *ssdeep* software**,**

Step 6: comparison of the values calculated by the Levenshtein method and using the *ssdeep* software,

Step 7:filtering the training datasets of neural networks from «noise» (the full implementation of this part of the algorithm is presented in [33]),

Step 8: training capsular neural network,

Step 9 training convolutional neural network,

Step 10 compute the Matthews correlation and resize the training datasets**.**

- $\phi = -1$ the received output data of both neural networks go beyond the value tolerance
- $\phi = 1$ the resulting outputs of both neural networks are correct (within the permissible deviation value)
- $\phi = 0$ the resulting output of both neural networks is random

Steps 11, 12: reconfiguring the training datasets and resizinge the CTPH.

Table 1 presents the results of calculating the value of CTPH and the editorial distance between the hashes of the obfuscated source code of mimikatz software using capsular, convolutional neural networks, as well as ssdeep software.

Table 2 shows the results of calculating the value of the context-piecewise hash of the obfuscated compiled source code and the editorial distance between the hashes of the mimikatz software using capsular, convolutional neural networks, and also the *ssdeep* software.

In the research, datasets used a comparison between files 20-40, 20-80, 20-128, 20-256, 20-512, 20-1024 bytes, as well as combinations of 40-512, 40-1024, 128-512, 128 -1024 bytes for *mimikatz, athena, engrat, grum, surtr, dyre* malware.

3. Results

Table 1.The results of computing the value of CTPH and the editorial distance between the hashes of the obfuscated source code of mimikatz software

File number in the dataset	mimikatz file hash values (20 byte)	mimikatz file hash values (512 byte)	Editorial distance	Percentage of malware samples calculated using scdeep	Percentage of malware samples computed using convolutional neural networks			Percentage of malware samples computed using capsule neural network		
					Training epoch			Training epoch		
					I	II	III	I	II	III
1.	b9be58b87140f922969c 905236829d2436c34400	ef73afe0b3862206e112 400dc97a6920c1240ca2	36	10	4	2	9	8	19	39
2.	e1077e747c9486dce1bf da820c078fe300a901fb	081cdfaf631a003a5a5d fa678b52af5c0eb2cbd3	36	13	6	8	7	16	27	42
3.	d86c9ca3861e333dc337 6fc5565943551389edd6	72840526d3cecbba084e ef91aed9c52cd94855d5	35	25	18	9	9	24	52	78
4.	bd72fda18edc004d5181 b57e48a757ac2ed94444	783e9520a25faca4f815 2dfc092d7d67e359c5f6	35	28	21	22	24	8	10	12
5.	8ab1d3267a46f953c73b 4154b1a261a8e02493d8	ad523321e582956d7b51 e9f4bc3763d9305231dc	30	11	3	7	3	34	54	82
6.	dc990c540fc50debf0cd c178101ab107acaef9fe	f2ba969ed8f8ecc7ce57 c54c39de5333cf0d6a8e	36	23	11	16	21	16	28	65
7.	b137df3d2083c226f985 c0494a9cef753034ac6d	f7fd9ed34bc6ead485bd 5e7c1b9f9f13f30fddba	34	13	10	9	12	16	27	46
8.	9efa06fa6567be9554db 5c351da39c9c084306e0	f7fd9ed34bc6ead485bd 5e7c1b9f9f13f30fddba	33	21	15	15	17	31	46	79
9.	4f5ec65628d2bde662a4 08854a41caea98c0f44f	f5cd09b85a44df103b21 ea9c4d02c564fcb19191	35	64	32	30	42	38	37	48
10.	5329b04a348368967844 f421453563001ad4ab89	37a56e3a4acbef542099 4c0d7864125e53f5aaa3	36	22	8	11	16	27	48	61
11.	95a56dfdfd7c8550afb8 ab2474916bb63e58bb15	37a56e3a4acbef542099 4c0d7864125e53f5aaa3	33	16	12	13	15	27	41	68
12.	aececb9dccd29fd5dd9 c0559ad62afb84af374b2	51168e0c2ab45361cf05 834a721cd4aba48098be	34	19	11	12	18	36	49	73
13.	14791ec8ec19ca534367 c54f008b8439eea89f09	497a16d6dd757f05fb88 4994c71bea880e87ad18	35	11	18	29	25	37	49	68
14.	dbfb0b8c0a28ea8bade 6306f9e8589ee1c310a39	c6ca0e98e0a66c45838f b254aec474553850ab91	34	16	14	21	29	52	58	71
15.	c91e176518b7e42450e2 c28d45bf31a1b3178240	7ad0cc0f4ba8c767fac7 f0a4f7ec192b3a60ec9e	36	18	16	19	28	29	43	68
16.	04b66940a08ac7adb0cd f19382a8169d0c256c09	5db88a72cdcfe90ff987 1eae5bf8d2b617d73b0a	37	26	11	19	36	39	56	73
17.	67b4a269a360b994d776 9e4b40220c8b59c219b0	fa926a049a1d9d72126b d07f1a1b87326b5e355b	34	41	27	11	29	26	58	61
18.	c2cdacd22e871ecef12b 0cbc8caf4559eecfa084	817c64fed50532e58dd2 1a8812c65fe10a250bd0	36	16	15	16	26	31	46	74
19.	4202fc70b1301ec50b1f 64ca525de6d31825787d	38bc177d79492834356f 1cce4f9120599f41e952	36	18	17	19	21	28	37	49
20.	20b5c47533cb97d72f9 0895ea1ffe27695063e54	818b59add29456248836 864d46c146d9d930d8a2	37	19	8	16	34	24	37	58

In training epochs 1-3, the results of the capsular neural network are better than the results of the convolutional neural network and ssdeep software, except for file №4 in the dataset, which is included in the statistical error.

Table 2. The results of calculating the value of CTPH and editorial distance between hashes of the compiled mimikatz source code.

File number in the dataset	mimikatz file hash values (20 byte)	mimikatz file hash values (512 byte)	Editorial distance	Percentage of malware samples calculated using ssdeep	Percentage of malware samples computed using convolutional neural networks			Percentage of malware samples computed using capsular neural network		
					Training epoch			Training epoch		
					I	II	III	I	II	III
1.	d7e4e9abedd0949b8bcf f30c7abbdad97b182be8	51f028f6b078f51583e0 a048d9bc577b6a4e17b9	37	25	23	31	42	17	19	23
2.	2c0e9d614fab60e18bd4 2e99659974a3d298a9ae	7f966e5a707dd69c13b5 de45c9765a9be437e642	35	16	18	14	22	8	11	9
3.	f76606cb6fae082991eb 271af5ab7629d592cb04	fb96549631c835eb239c d614cc6b5cb7d295121a	32	28	27	36	45	16	17	14
4.	14da593832768f0a08e8 ecd46363936eef096dcc	72ac7a00a3c2a0a825cd 016d71b0d587c6cc3f46	36	23	16	22	34	18	20	16
5.	7f01a23afa1bcecdfdbb 25b953c4f15366eaba51	35139ef894b28b73bea0 22755166a23933c7d9cb	37	37	34	41	48	27	29	23
6.	1ca12a53c82cdd508054 bdcdbe5256ccdd44c13c	918b1c05e576f4b90fce 15a06bc3442d72852a3c	35	48	44	53	61	34	31	28
7.	a7f0499bf3eb6180d4da 748426822404e46dea13	4759f2ba1ba20f493664 dbf5e36c1a1ec0d75658	36	15	11	13	12	8	2	3
8.	aec2a4accb7ca456a57a c4426e8f51c2e6a8b143	902a2d132f213700b5de fbefe7567f68ca8e234a	35	19	18	26	29	16	10	13
9.	582d2ceff8f4f493f3a9 d45c71286255946a7d37	b2fd9a1405ba74fc360e 1784961176b2b88bf5c9	37	39	28	48	57	25	23	12
10.	a25a87930b155282e138 35142ad63cea1994d02d	c47419fdd4d6f146e430 64b9ddb859a250404500	36	53	47	40	57	34	29	47
11.	2f7b14912dddcf7c1c7a ebb49955cb5bf0ab3257	b521d7652866027a7e5b 43c6269d7c81ffb5a86e	36	28	30	37	44	14	19	23
12.	fd5fd2f7953cf5630f74 c2933b378d4381367ddd	9de4bfa1fdb6c90637d3 5492ec14ee10a3967997	33	56	49	53	67	42	48	34
13.	e88dac72cd8ac64360d9 5fb15e8ea9aaa8794f8c	1eb796fd1ff7dda036fc a37d0f31aab19dedab1a	37	24	29	48	52	17	23	15
14.	efa91cc773ee2c32ba51 2ffce8db8a3760bda564	99828f68be57c53ff954 5f79e32bdb36050bf93b	32	19	27	29	37	13	18	28
15.	f9980d6122acf1bf54a6 8e49d15507fbc3ce7c1f	2400b40333821b00b5d0 b67f20f5f0e30ebf02dd	36	56	44	58	63	37	34	39
16.	c5d4d95ce32029e1150a 20d2f836b7b2c6e49546	dfb380d8b0709104c606 978092c7164160f32887	37	29	27	35	38	21	17	34
17.	5156507d0b07bd9eaafe 56815e1a04a0eaa1a8e9	bd951f174a8f0f211c62 bc1869d69f581788ee59	37	48	27	44	56	25	38	14
18.	14fd3fa5756432336c73 656c76f4751aa6f707f9	b9acd4446a9ee133799f a3d8f3e35e001c616776	37	16	24	36	38	8	10	11
19.	f1d8238c9141f46246bf 2193908b1be6f87b09f8	f1513655d577bf56bcf86 2b1851e66bb683d373c	33	56	48	56	61	32	27	46
20.	50effcaad368f00bfc71 2105a708ff917f9f95d0	49a48ed249c7b82959aa 85b9470938bbcc9c45cc	36	36	27	38	46	16	28	31

In epochs 1-3 of training for compiled software, the results of the capsule neural network are worse worse than the results of the convolutional neural network and ssdeep software.

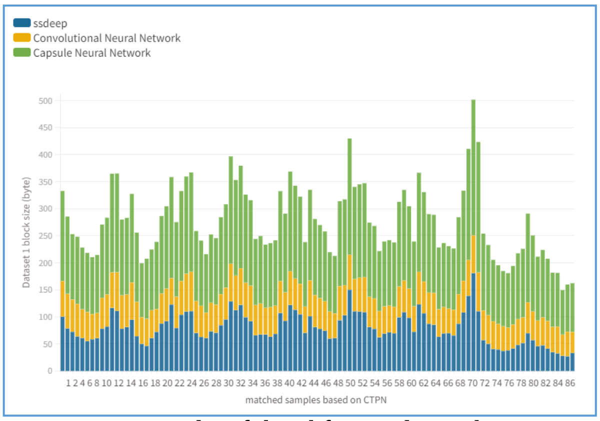


Fig. 4. CTPH results of the obfuscated mimikatz source code.

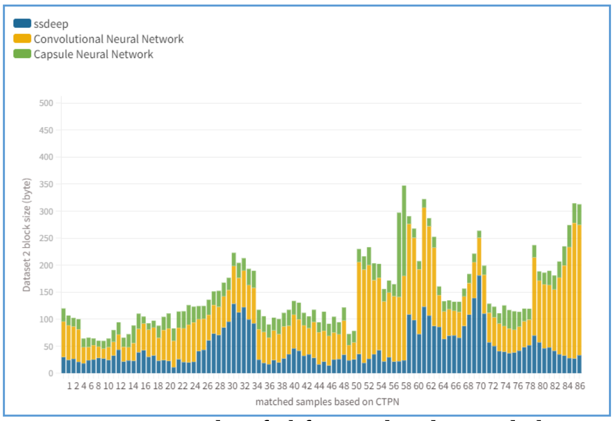


Fig. 5. CTPH results of obfuscated and compiled mimikatz source code.

The use of a convolutional neural network is not always justified, since the degree of detection is comparable to the degree of detection by *ssdeep* software. The use of a capsule neural network for malware detection is justified in the presence of the source code (even in an obfuscated state), since even after the first training epoch, the detection results are not worse (and in most cases better) than the detection results using ssdeep and a trained convolutional neural network.

Tables 3 and 4 present the results of the studies of the operation of capsule and convolutional neural networks, based on datasets obtained from the *obfuscated mimikatz source code* with three training epochs and a variable block size of CTPH.

Table 3. Number of detected threats.

Number of datasets (dataset 1)	Number of datasets (dataset 2)	The number of samples detected and classified as threats on different sizes (20, 40, 128 bytes) and three epochs (I, II, III) of training by a capsule neural network									The number of samples detected and classified as a threat at different sizes (20, 40, 128 bytes) and three epochs (I, II, III) of training by a convolutional neural network									Number of detected but mismatched malware samples *		
CTPN size (byte)		20			40			128			20			40			128					
Training epoch		I	II	III	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III
100	100	7	7	9	11	13	12	12	15	18	3	3	4	4	6	6	9	10	1	-	-	1
200	200	10	11	11	12	14	16	17	17	21	5	4	6	6	8	5	8	5	6	-	1	2
300	300	12	12	14	16	18	23	28	29	22	8	7	8	8	9	11	13	15	16	1	1	2
350	350	12	13	15	15	16	18	21	26	25	7	7	11	10	12	18	16	18	19	2	2	3
450	450	14	16	19	19	22	26	29	34	38	10	9	11	12	16	18	18	21	20	2	1	4
500	500	14	16	18	19	21	27	29	33	36	11	10	13	16	15	15	17	19	19	2	2	4
600	600	22	25	29	30	34	35	39	41	44	14	15	11	19	24	26	20	25	26	3	3	3
800	800	37	41	46	48	52	55	57	57	60	22	26	27	29	34	37	39	44	45	5	4	6
950	950	42	42	46	47	58	60	66	68	68	28	29	28	31	33	39	42	46	49	4	4	4
1000	1000	42	43	47	50	51	59	61	65	69	34	33	35	30	35	39	49	52	55	5	6	3

*The number of detected but mismatched malware samples separately detected by both neural networks. These samples were output to a special dataset and verified by publicly available malware detection resources.

Table 4. Number of detected threats.

Number of datasets (dataset 1)	Number of datasets (dataset 2)	The number of samples detected and classified as threats at different sizes (256, 512, 1024 bytes) and three epochs (I, II, III) of training by a capsular neural network									The number of samples detected and classified as a threat at different sizes (20, 40, 128 bytes) and three epochs (I, II, III) of training by a convolutional neural network									Number of detected but mismatched malware samples *		
CTPN size (byte)		256			512			1024			256			512			1024					
Training epoch		I	II	III	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III
100	100	18	14	16	14	16	19	8	12	14	7	11	14	9	11	14	7	8	11	-	1	1
200	200	18	12	12	14	18	19	11	13	10	3	4	3	5	8	11	5	9	14	1	1	2
300	300	17	19	16	14	17	12	10	21	23	9	11	10	8	12	9	8	8	13	-	2	2
350	350	18	18	21	18	21	23	23	27	27	9	15	17	12	18	14	14	11	12	2	2	3
450	450	22	26	28	29	29	34	20	23	25	12	15	13	20	16	16	17	29	13	2	5	3
500	500	23	24	29	31	33	30	28	21	32	16	12	15	22	22	25	28	26	25	3	7	7
600	600	28	31	30	32	35	39	34	38	41	20	24	21	24	28	25	29	34	31	5	6	6
800	800	37	37	39	41	46	39	42	46	49	31	28	34	34	25	27	39	32	34	7	9	11
950	950	48	53	53	52	58	56	64	65	56	34	30	31	35	38	38	39	42	45	11	9	10
1000	1000	47	52	51	56	61	60	64	66	68	40	42	46	42	44	44	47	49	51	8	11	12

Fig. 6 shows a report from the *virustotal* service when examining one of the *mimikatz* malware samples detected by neural networks. In particular, the virustotal service did not detect either the file type or whether CTPH (based on ssdeep) belongs to a particular type of malware.

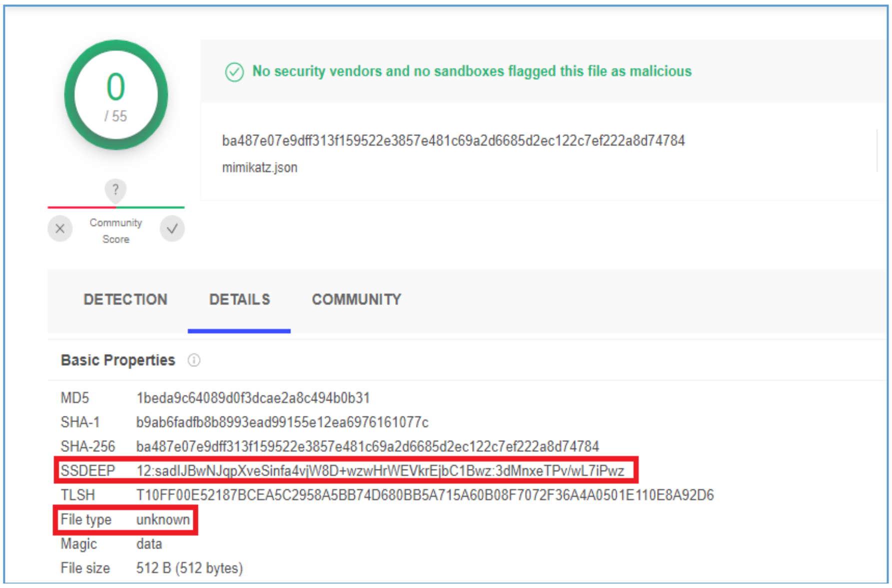


Fig. 6.Virustotal service report.

Tables 5 and 6 present the results of the studies of the operation of capsule and convolutional neural networks, based on data sets from the *obfuscated compiled code* of the *mimikatz* software.

Table 5. Number of detected threats.

Number of datasets (dataset 1)	Number of datasets (dataset 2)	The number of samples detected and classified as threats at different sizes (20, 40, 128 bytes) and three epochs (I, II, III) of training by a capsular neural network									The number of samples detected and classified as a threat at different sizes (20, 40, 128 bytes) and three epochs (I, II, III) of training by a convolutional neural network									Number of detected but mismatched malware samples *		
CTPN size (byte)		20			40			128			20			40			128					
Training epoch		I	II	III	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III
100	100	2	1	2	3	2	3	3	4	4	2	2	3	3	3	4	5	2	3	-	-	-
200	200	3	2	3	3	4	2	2	3	3	1	1	2	2	3	2	4	3	4	-	-	-
300	300	3	4	4	4	4	5	3	5	5	2	3	3	4	3	4	4	4	4	-	-	1
350	350	3	3	4	4	5	5	5	6	6	3	3	3	3	4	5	5	4	4	-	1	1
450	450	4	5	5	5	6	6	6	8	9	3	4	4	4	5	6	5	7	7	-	1	-
500	500	3	5	5	5	6	8	8	9	11	4	4	5	5	7	9	9	10	10	-	2	2
600	600	5	6	6	6	8	9	11	11	12	5	4	7	7	9	11	10	11	10	1	1	1
800	800	7	6	7	7	8	11	13	14	14	6	8	9	8	8	9	8	11	13	2	1	2
950	950	9	9	10	11	9	11	12	15	15	8	10	10	11	13	15	14	15	17	2	2	3
1000	1000	11	13	14	14	14	15	17	19	18	10	11	11	11	13	16	18	21	23	2	4	4

Table 6. Number of detected threats

Number of datasets (dataset 1)	Number of datasets (dataset 2)	The number of samples detected and classified as threats at different sizes (256, 512, 1024 bytes) and three epochs (I, II, III) of training by a capsular neural network									The number of samples detected and classified as a threat at different sizes (256, 512, 1024 bytes) and three epochs (I, II, III) of training by a convolutional neural network									Number of detected but mismatched malware samples *		
CTPN size (byte)		256			512			1024			256			512			1024					
Training epoch		I	II	III	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III
100	100	9	11	12	12	14	14	15	16	16	8	8	10	11	13	12	11	11	10	-	-	1
200	200	10	12	13	14	13	13	15	15	12	11	10	11	12	11	13	12	13	14	-	1	1
300	300	11	12	12	15	17	18	19	18	18	10	12	13	12	14	14	15	14	14	-	-	-
350	350	11	11	12	12	12	16	15	11	14	14	12	13	15	15	15	18	19	21	-	1	2
450	450	13	12	13	13	15	15	16	17	18	11	12	13	14	16	16	15	17	19	2	3	3
500	500	12	14	14	14	15	14	15	11	12	11	10	11	13	14	12	15	15	16	-	1	2
600	600	10	11	12	10	12	12	12	14	13	9	10	11	12	10	10	14	15	14	1	2	2
800	800	12	14	15	15	16	17	17	18	18	16	14	15	15	16	17	18	21	19	2	3	3
950	950	12	13	12	14	15	15	16	18	19	12	12	13	14	15	16	12	15	16	2	3	4
1000	1000	12	12	13	13	15	16	16	17	18	11	10	12	15	16	17	18	19	20	2	2	3

Given the malware source code (or fragment), the capsule neural network performs better than the convolutional neural network in detecting obfuscated malware. But when compiled, the detection performance of the capsular neural network decreases. Also, both neural networks separately detected a small set of data and software fragments classified as malware. Figures. [7]-[12] show a visualization of the output data of a capsule neural network with 3 training epochs and CTPN datasets, 20, 40, 80, 128, 256, 512 bytes.

Fig. 7. Visualization of malware detection results by capsule neural network. (I training epoch, CTPH size 20 bytes)

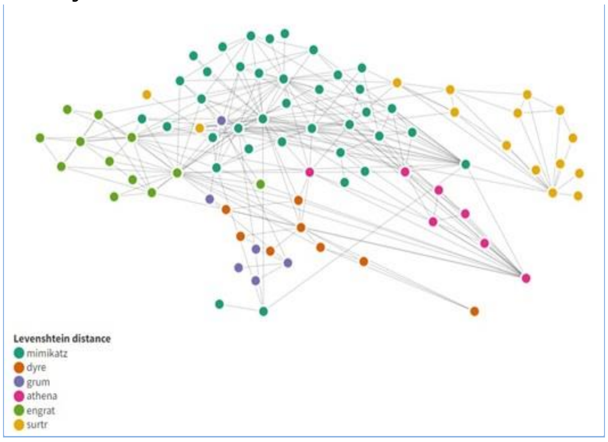


Fig. 8. Visualization of malware detection by capsule neural network. (I training epoch, CTPH size 40 bytes)

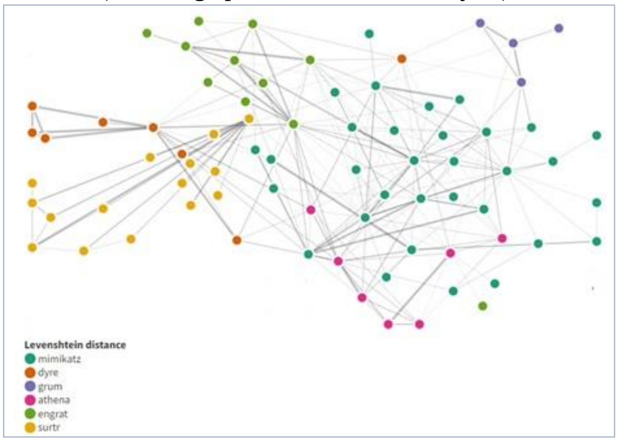


Fig. 9. Visualization of malware detection results by capsule neural network. (II training epoch, CTPH size 80 bytes)

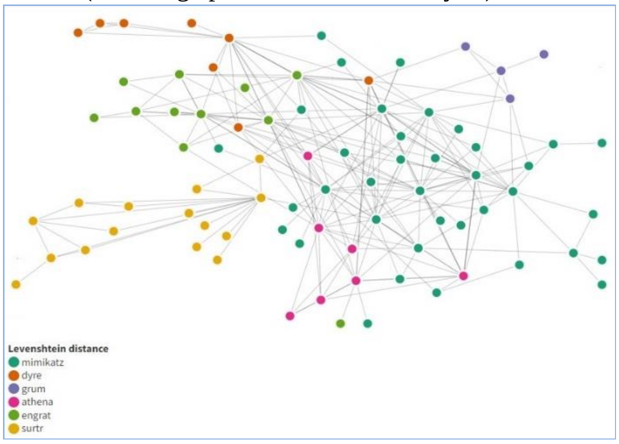


Fig. 10. Visualization of malware detection by capsule neural network. (II training epoch, CTPH size 128 bytes)

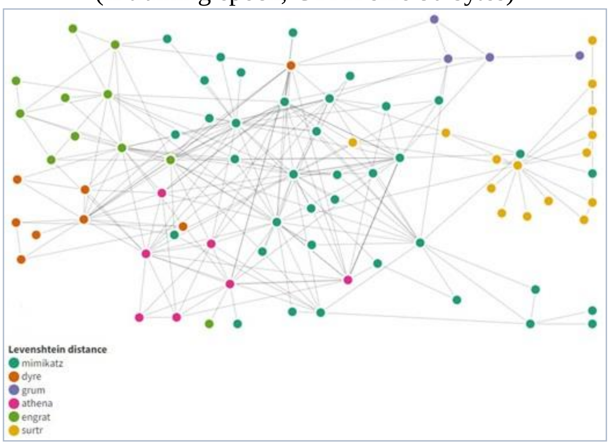


Fig. 11. Visualization of malware detection results by capsule neural network. (III training epoch, CTPH size 256 bytes)

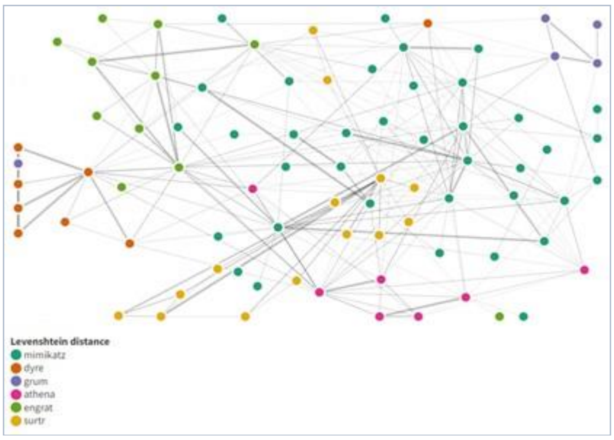


Fig. 12. Visualization of malware detection results by capsule neural network. (III training epoch, CTPH size 512 bytes)

With an increase in the size of the CTPH files (interval 256, 512, 1024 bytes) for training the capsule network, the increase in the detection of the number of malware code fragments is insignificant (0.3-0.5%, Fig. 7, Fig. 8, Table 6) in contrast to files 20 , 40, 128 bytes (12-14% increase). But increasing the size of the CTPH file allows increasing the editorial distance (Figure 9-12) to granularly group malware by type.

4. Conclusion

This paper proposes the use of transfer learning of a capsule neural network to detect obfuscated malware. Convolutional and capsule neural networks were trained on the same datasets. The source codes of *mimikatz, athena, engrat, grum, surtr, dyre* malware were used as datasets. When building an intrusion detection system using neural networks, their complex application is necessary. Annotated malware datasets are critical when training neural networks. The use of transfer learning of a capsule neural network to detect malware is justified if the source code of the malware or its fragments (preferably the first versions) is available. In this case, the neural network detects malware, even with its high degree of obfuscation. But in the absence of source code, the effectiveness drops, yielding to «standard» means of detecting malware. The use of the CTPH method for generating «weight» coefficients of a neural network is most effective with a small file size of CTPH.
Increasing the editorial distance increases the selectivity of detecting different types of malware.

References

[1] D. Ashok Kumar and S. R.Venugopalan, “Intrusion Detection Systems: A Review” *International Journal of Advanced Research in Computer Science,* vol. 8, no 8, pp.356--370, 2017.

[2] O. Shelukhin, D. Sakalema and A.Filinov, *Detection of intrusions into computer networks.* Hot line-Telecom, 2018.

[3] S. Survey and D. Usha, “A survey of intrusion detection system in IoT devices”, *International Journal of Advanced Research (IJAR),* vol 6, pp. 23-31, 2018.

[4] H.Hindy et al., “A taxonomy of network threats and the effect of current datasets on intrusion detection system”, *arXiv preprint arXiv:1806.03517,* 2020.

[5] Tuan-Hong Chua and Iftektar Salam, “Evaluation of machine learning algorithms in network-based intrusion detection system”, *arXiv preprint arXiv:2203.05232,* 2022.

[6] Snort intrusion detection and prevention system official website. [Online]. Available https://www.snort.org/

[7] Suricata intrusion detection and prevention system official website. [Online]. Available https://suricata.io/

[8] Zeek an open source Network Security Monitoring tool system official website. [Online]. Available https://zeek.org/

[9] Cisco NGIPS system web pages. [Online]. Available https://www.cisco.com/c/ru_ru/products/security/ngips/index.html

[10] F.Maymi and S.Harrris, *CISSP, Exam Guide,* Ninth Edition, Mc Graw Hill, New York, San Prancisco, Singapore, Sydney, Toronto, 2022.

[11] C. Chio and D. Freeman, *Machine Learning and Security,* O`Reilly® , Boston•Sebastopol• Tokyo, 2020.

[12] M. Collins, *Network Security. Through Data Analysis,* O`Reilly® (DMK press), 2020.
[13] ISO/IEC 7498-1, Second edition 1994-11-15. Corrected and reprinted, 1996.
[14] MITRE ATT&CK® official website. [Online]. Available https://attack.mitre.org/matrices/enterprise/
[15] CVE cybersecurity web pages. [Online]. Available https://cve.mitre.org/index.html
[16] OWASP Cheat Sheet Series. [Online].Available https://cheatsheetseries.owasp.org/cheatsheets/Attack_Surface_Analysis_Cheat_Sheet.html
[17] A. Cheremushkin, *"Cryptographig protocols: Main properties and vulnerabilites*", PDM , vol.2 appendix, pp.115-150, 2009.
[18] T. V. Jamgharyan and V.H.Ispiryan, "Model of generative network attack" *Proceedings of 13th International Conference on Computer Science and Information Technologies (CSIT),* Yerevan, Armenia, pp. 90-94, 2021.
[19] A. Ul Haq et al, "Addressing tactic volatility in self-adaptive systems using evolved recurrent neural networks and uncertainty reductions tactics", *arXiv preprint arXiv:2204.10308v1,* 2022.
[20] S. Das, "FGAN: Federated generative adversarial networks for anomaly detection in network traffic", *arXiv preprint arXiv:2203.11106v1,* 2022.
[21] Sk.Tanzir Mehedi, "Dependable intrusion detection system for iot: a deep transfer learning –based approach", *arXiv preprint arXiv:2204.0483v1,*2022.
[22] I. Panagiotis et al, "Securing the Smart Grid: A Comprehensive Compilation of Intrusion Detection and Prevention Systems", *DOI 10.1109/Access,* 2017.
[23] A. S. Dina et al , "Effect of balancing data using synhthetic data on the performance machine learning classifiers for intrusion detection in computer networks", *arXiv preprint arXiv:2204.00144v1,*2022.
[24] T.Nathuya and G.Suseendram, *An Effective Hybrid Intrusion Detection System for Use in Security Monitoring in the Virtual Network Layer of Cloud Computing Technology,* Springer Nature, Singapore, 2019.
[25] E.Pelofske, "A robust cubersecurity topic classification tool", *International Journal of Network Security & Its Application (IJNSA),* vol.14, № 1, pp. 1-25, 2022.
[26] G.Renjith et al, "GANG-MAM: GAN based enGine for modifying android malware" *arXiv preprint arXiv: 2109.13297,* 2021.
[27] F.Zhong et al, "MalFox: Camouflaged adversarial malware example generation based on Conv-GANs againist black—box detectors", *arXiv preprint arXiv: 2011.01509,* 2021.
[28] B.E.Zolbayar et al, "Generating practical adversarial network traffic flows using NIDSGAN", *arXiv preprint arXiv: 2203.06694v1,* 2022.
[29] Md.Ariful Haqua, R.Palit, " A review on deep neural network for computer network traffic classification", *arXiv preprint arXiv: 2205.10830v1,* 2022.
[30] D. Kus et al, "A false sense of security? Revisting the state of machine learning-based industrial intrusion system", *arXiv preprint arXiv: 2205.09199v1,* 2022.
[31] S.Layeghy and M. Portmann, "On generalisibility of mashnine learning-based network intrusion detection systems", *arXiv preprint arXiv: 2205.041112v1,*2022.
[32] S.Sohail et al, "Explainable and optimally configured artifical neural networks for attack detections in smart homes", *arXiv preprint arXiv:2205.080443v1,*2202.
[33] T. Jamgharyan, "Research of the data preparation algorithm for training generative-adversarial network", Bulletin of High Technology, no. 19, pp. 40-50, 2022.
[34] Kaggle datasets base website. [Online]. Available https://www.kaggle.com/datasets
[35] Registry of Open Data on AWS website. [Online]. Available

https://registry.opendata.aws/

[36] Public data sets for testing and prototyping. [Online]. Available https://docs.microsoft.com/en-us/azure/azure-sql/public-data-sets?view=azuresql

[37] Datasets base website. [Online]. Available http://apolloscape.auto/

[38] Datasets of overhead imagery. [Online]. Available http://xviewdataset.org/#dataset

[39] Google open images dataset. [Online]. Available https://ai.googleblog.com/2016/09/introducing-open-images-dataset.html

[40] MalwareBazaar Database. [Online]. Available https://bazaar.abuse.ch/browse/

[41] Malware database. [Online]. Available http://vxvault.net/ViriList.php

[42] A free malware repository for researches. [Online]. Available https://malshare.com/

[43] Malware repository. [Online]. Available https://avcaesar.malware.lu/

[44] Malware repository. [Online]. Available https://www.virusign.com/

[45] Viruses repository. [Online]. Available https://virusshare.com/

[46] A live malware repository. [Online]. Available https://github.com/ytisf/theZoo

[47] F.Wang et al, "An efficient unsupervised domain adaptation deep learning model for unknown malware detection", *International conference on security and privacy in new computing environments (SPNCE),* vol. 423, pp. 64 -76, 2022.

[48] G. Pitolli et al, "MalFamAware: automatic family identification and malware classification through online clustering", *International Journal of information security* vol. 20, pp. 371-386, 2021.

[49] S. David, R. Anand, V. Jeyakrishnan and M Niranjanamurthy, *"Security issues and privacy concerns in industry 4.0 applications",* Wiley, Beverly, 2021.

[50] I. Priyadarshimi and R.Sharma, *"Artifical Intelligency and Cybersecurity",* CRC Press Taylor&Francis Group, New York, 2022.

[51] Encyclopedia by Kasperky. [Online].Available https://encyclopedia.kaspersky.ru/glossary/indicator-of-compromise-ioc/

[52] Nettitude labs web site. [Online].Available https://labs.nettitude.com/blog/context-triggered-piecewise-hashing-to-detect-malware-similarity/

[53] S.Kumar and Sudhakar, "*MCFT-CNN: Malware classification with-tune convolutional neural networks using traditional and transfer learning in IoT",* DOI 10.1016 Future Generation Computer systems, vol.25 pp. 334-351, 2021.

[54] C.Rong et al, "TransNet: Unseen malware variants detection using deep transfer learning", International Conference on Security and Privacy in communication systems (LNICST) vol.336, pp.84-101, 2020.

[55] R.Mortier et al, "Distributed data analysis", *arXiv preprint arXiv:*.2203.14088.2021.

[56] D.Pogorelov et al, "Comparative analysis of the Levenstein and Dameray-Levenstein edit distance algorithms", *Processing of Moscow State University after N.Bauman,* vol. 31, pp. 803-811, 2019.

[57] ssdeep software project website. [Online].Available https://ssdeep-project.github.io/ssdeep/index.html

[58] Professional information and analytical resource dedicated to machine learning, pattern recognition and data mining. [Online].Available http://www.machinelearning.ru/wiki/index.php?title=%D0%9A%D0%BE%D1%80%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D1%8F_%D0%9C%D1%8D%D1%82%D1%8C%D1%8E%D1%81%D0%B0

[59] Capsule networks paperspace. [Online]. Available https://blog.paperspace.com/capsule- networks/

[60] Free service that analyzes malware. [Online].Available https://www.virustotal.com/
[61] Malware scanning platform. [Online].Available https://www.herdprotect.com/
[62] "Dotfuscator" software web pages. [Online].Available https://docs.microsoft.com/ru-ru/visualstudio/ide/dotfuscator/capabilities?view=vs-2022
[63] "Guardsquare" software web site. [Online]. Available https://www.guardsquare.com/proguard

Կապսուլային նեյրոնային ցանցով օբֆուսկացված վնասաբեր ծրագրային ապահովման հետազոտում

Թիմուր Վ. Ջամղարյան

Հայաստանի ազգային պոլիտեխնիկական համալսարան
e-mail: t.jamgharyan@yandex.ru

Ամփոփում

Ներխուժման հայտնաբերման և կանխարգելման համակարգերը ցանցային ենթակառուցվածքի անվտանգության ապահովման անբաժանելի բաղադրիչն են: «Դասական» ներխուժման հայտնաբերման և կանխարգելման համակարգերը չեն կարողանում հայտնաբերել այնպիսի սպառնալիքներ, որոնք նկարագրված չեն համակարգի կանոններում: Բացի այդ, նաև բաց խնդիր է համարվում օբֆուսկացիայի ենթարկված վնասաբեր ծրագրային ապահովման հայտնաբերումը:
Ծրագրային ապահովման և ցանցային ենթակառուցվածքի անվտանգությունով զբաղվող հետազոտողները, փորձում են նշված խնդիրը լուծել մեքենայական ուսուցման միջոցով: Հետազոտությունում ներկայացված են փոխանցման ուսուցման մեթոդով ուսուցանված կապսուլային նեյրոնային ցանցի ցուցաբերած արդյունքները վնասաբեր ծրագրային ապահովման հայտնաբերելու հարցում: Հետազոտությունը իրականացվել է վնասաբեր ծրագրային ապահովման ելակետային կոդի հիման վրա, կիրառելով համատեքստա-մասնատված հեշավորման մեթոդը: Վնասաբեր ծրագրային ապահովման ելակետային կոդերը ստացվել են հանրահասանելի աղբյուրներից: Կապսուլային նեյրոնային ցանցի ուսումնասիրության արդյունքները համեմատվել են նախապես ուսուցանված փաթույթային նեյրոնային ցանցի և վնասաբեր ծրագրային ապահովման հայտնաբերելու հանրահասանելի համացանցային ծառայությունների միջոցով: Մշակված ծրագրային ապահովման ելակետային կոդերը, նախապես ուսուցանված մոդելը, տվյալների հավաքածուների մի մասը, հոդվածում չներառված հետազոտության արդյունքները հասանելի են https://github.com/T-JN կայքում:

Բանալի բառեր՝ կապսուլային նեյրոնային ցանց, անորոշ հեշավորում, ներխուժման հայտնաբերման համակարգ, խմբագրական հեռավորույուն, ցանցային ենթակառուցվածք:

Исследование обфусцированного вредоносного программного обеспечения с помощью капсульной нейронной сети

Тимур В. Джамгарян

Национальный политехнический университет Армении
e-mail: t.jamgharyan@yandex.ru

Аннотация

Системы обнаружения и предотвращения вторжений являются неотьемлимым компонентом безопасности сетевой Инфраструктуры. Классические системы обнаружения и предотвращения вторжений не в состоянии обнаружить угрозу не описанную в наборе правил. Также нерешенной полностью задачей является: задача обнаружения вредоносного программного обеспечения подвергнутого обфускации.
Исследователи в сфере безопасности программного обеспечения и сетевой Инфраструктуры пытаются решить данные задачи с помощью машинного обучения.
В работе представлены результаты исследования использования трансферного обучения капсульной нейронной сети для обнаружения вредоносного программного обеспечения. Исследование проводилось на основе исходного кода вредоносного программного обеспечения с использованием метода контекстно-кусочного хеширования. Исходные коды вредоносного программного обеспечения были получены из общедоступных источников программного обеспечения. Проверка результатов обучения капсульной нейронной сети проводилась с использованием обученной сверточной нейронной сети и общедоступных источников тестирования вредоносного программного обеспечения. Исходные коды разработанного программного обеспечения, часть наборов данных для обучения нейросети, результаты исследования не внесенные в статью представлены по адресу https://github.com/T-JN

Ключевые слова: капсульная нейронная сеть, нечеткое хэширование, система обнаружения вторжений, редакционное расстояние, трансферное обучение.

Mathematical Problems of Computer Science 58, 84–90, 2022.
doi: 10.51408/1963-0095

UDC 004.62:004.946

Data Processing and Persistence in Virtual Reality Systems

Arman A. Hovhannisyan

National Polytechnic University of Armenia
e-mail: aahovhannisyan1@gmail.com

Abstract

Data processing and persistence are key aspects of developing a Virtual Reality system. In this paper, an improvement is offered to the distance calculation algorithm of the Unity Engine. Additionally, data persistence mechanisms provided by the Unity Engine are reviewed, and File System is selected as an appropriate option. Storage of object coordinates to the File System is implemented. The results provide a baseline for developing a system for creating virtual stands for professional research.

Keywords: Virtual reality, Data management, File System, Serialization.
Article info: Received 29 March 2022; received in revised form 9 October 2022; accepted 17 November 2022.

1. Introduction

In virtual reality, data is represented both in primitive types (int, float, string) and complex types provided by the engine. Object positions in space are determined in the Cartesian coordinate system [1] (see Fig. 1).

A common task is to compare the distance of 2 points from a given point $A(x_1, y_{1,} z_1)$. The Unity Engine Scripting API [2] provides a complex data type called Vector3 to store object coordinates, along with its Vector3.Distance() method to calculate distance between 2 points. Given the points (x_2, y_2, z_2), $C(x_3, y_3, z_3)$, this method may be used to accomplish the task, comparing the following values: Vector3.Distance(B, A), Vector3.Distance(C, A).
A more efficient solution may be applied using the formula of the distance between 2 points [1]:

$$d_{AB} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}, \tag{1}$$

$$d_{AC} = \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2} \quad (2)$$

Instead of comparing values for d_{AB} and d_{AC}, the radicands may be compared, saving CPU time on unnecessary calculations.

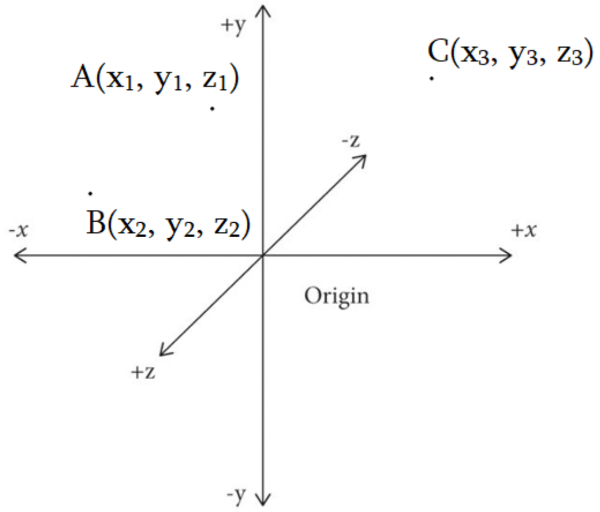


Fig. 1. Object positions in space.

To have persistent data between sessions, the user progress has to be stored on the disk. There are several methods of managing data storage, including SQL database, PlayerPrefs and OS File System. On specific events during the runtime, which are to be defined, data containing all the current values have to be stored. These events may include user interaction, object state mutation, or events may be set to trigger on specific timestamps, e.g., every 10 seconds. Then, on the next program run, these stored values have to be fetched and transmitted to the engine to render the objects in the same state and position, as they were when the last event was triggered.

2. Persistent Data

To have persistent data between sessions, the user progress has to be stored on the disk. Below are listed several methods of managing data storage.

1. SQL Database

SQL is useful when there is relational data. It supports queries to fetch related data sets. In our case, we have just objects that need to be memorized and then retrieved on the next run. Such simple operations are easier to implement and faster in work on File System. SQL is a dedicated software and isn't an integrated part of Operating Systems, as File System is. Also, a connection to SQL service should be kept active during the runtime.

2. PlayerPrefs

PlayerPrefs is a class provided by Unity Engine that stores Player preferences between game sessions. It stores values in the OS registry. Though it is possible to store data using this method, it is not recommended to do so. This method should be used for data, that can be afforded to lose, such as user settings and preferences. Sensitive and relatively big data should not be kept in registries.

3. File System

To store data in files, it needs to be formatted in some way. It may be serialized [3] to binary format and written to a file. That data will then be successfully deserialized and used in the application. But since binary is not human-readable, it makes this format insufficient. Moreover, it is not possible to edit the saved data manually. Using JSON data type allows bypassing these problems.

Taking into account the points mentioned above, it was decided to handle data storage using File system and Serialization, so every time data needs to be stored, it is serialized to JSON format and written to a file (see Fig. 2). Then, to restore the state in the application, the file is read and data is deserialized to object (see Fig. 3).

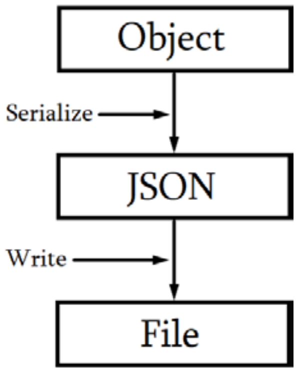


Fig. 2. Storing Data.

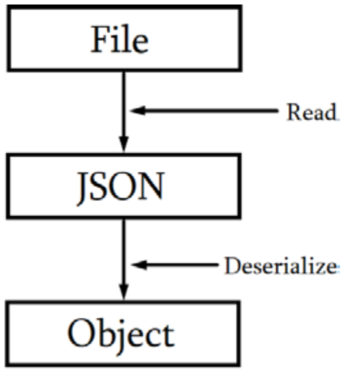


Fig. 3. Using Stored Data.

3. Saving Object Position

A specific and common example is persisting the object position. In this example, we have a cube placed on a table (see Fig. 4). The Origin (0, 0, 0) can be located on the ground.

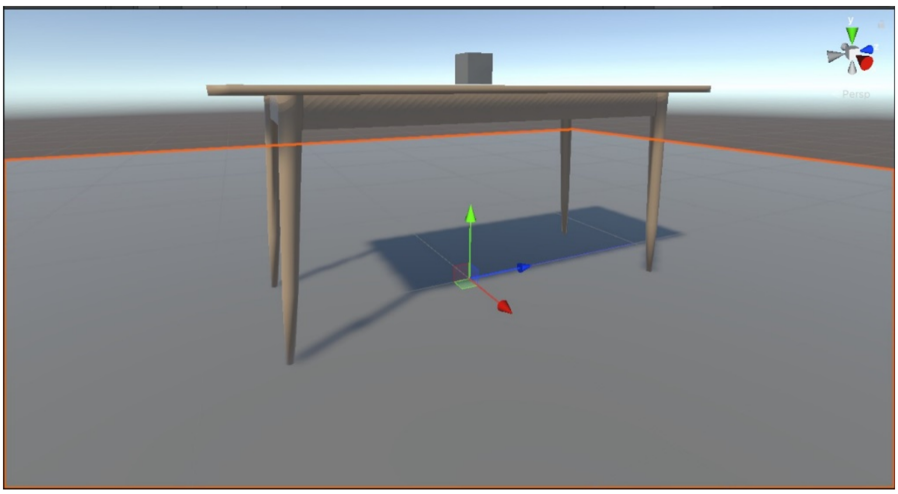


Fig. 4. Cube.

To save the cube position after it is replaced, a class called SaveManager is created, which contains 2 methods: save and load. These methods use a file called "position.dat" to write/read data.

SaveManager.cs

```
public class SaveManager
{
    public static void save(Vector3 pos)
    {
        string path = Path.Combine(Application.persistentDataPath, "position.dat");
        File.WriteAllText(path, JsonUtility.ToJson(pos));
    }

    public static Vector3 load()
    {
        string path = Path.Combine(Application.persistentDataPath, "position.dat");
        string result = File.ReadAllText(path);
        Vector3 pos = JsonUtility.FromJson<Vector3>(result);
        return pos;
    }
}
```

The save and load methods would then be invoked from a script, which is bound to the object. The save method would be bound to the XR Grab Interactable component [4] “Select Exited” event to save data every time the object is released. The load method would be invoked from the Start method to set object positions from the saved data on a fresh program run.

CubeScript.cs

```
public class CubeScript : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
        Vector3 position = SaveManager.load();
        transform.position = position;
    }

    public void Save()
    {
        SaveManager.save(transform.position);
    }
}
```

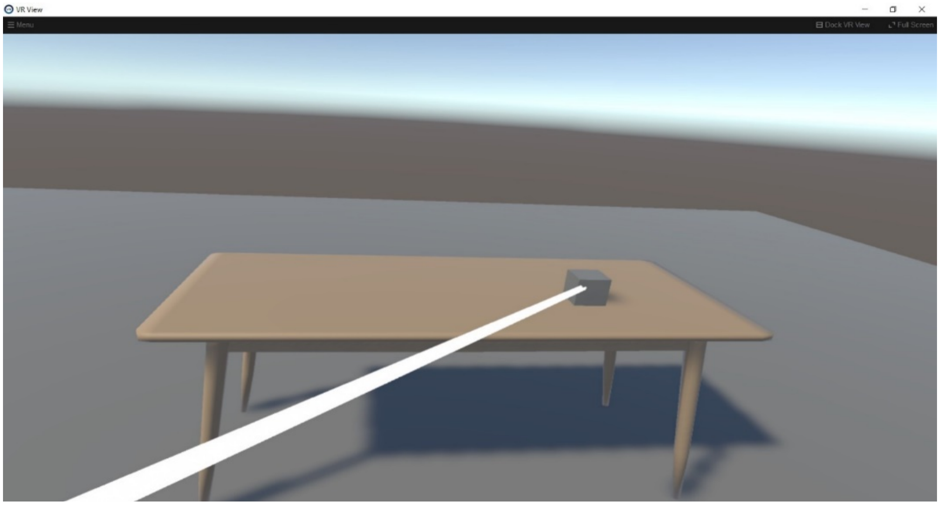


Fig. 5. Cube position changed via left controller.

The saved file position.dat:
{"x":-0.0030583017505705358,"y":0.838373601436615,"z":-2.0934112071990969}
After restarting the program, we still have the cube in its new place (see Fig. 5).
Now we can modify this file content, and set the coordinates to (0, 0, 0).
{"x":0,"y":0,"z":0}
After modifying and saving the file, and running the program again, we can see that the cube appears on the Origin as expected (see Fig. 6).

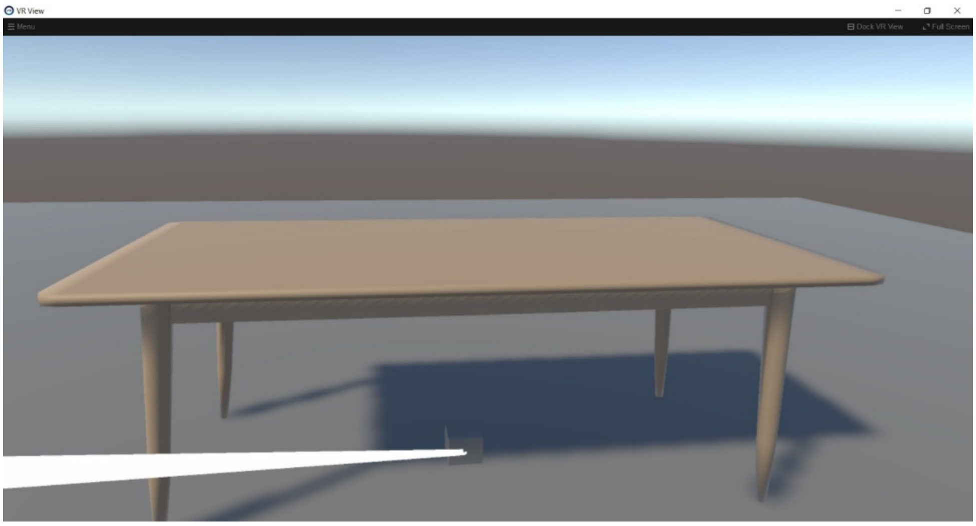


Fig. 6. Cube position manually set to Origin.

4. Conclusion

In this article, an improvement to the Unity Engine distance calculation algorithm was suggested. Additionally, data types provided by the Unity Engine were reviewed. Data storage options were compared and decided to use the OS File System and data serialization. As an example, a cube position storage and loading were implemented. This method will be used also for custom complex data types to store, marking the class representing the data type as Serializable.

References

[1] Wikipedia, (2012) Cartesian Coordinate System. [Online]. Available: https://en.wikipedia.org/wiki/Cartesian_coordinate_system

[2] Unity Engine Scripting API Reference. [Online]. Available: https://docs.unity3d.com/ScriptReference/

[3] Microsoft Docs, (2021) Serialization (C#). [Online]. Available: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/serialization/

[4] The Khronos Group Inc., “The OpenXR Specification”.

[5] Unity Learning, (2020) Create with VR. [Online]. Available: https://learn.unity.com/course/create-with-vr?uv=2020.3

Տվյալների մշակումը և պահպանումը վիրտուալ իրականության համակարգերում

Արման Ա. Հովհաննիսյան
Հայաստանի ազգային պոլիտեխնիկական համալսարան
e-mail: aahovhannisyan1@gmail.com

Ամփոփում

Տվյալների մշակումը և պահպանումը վիրտուալ իրականության համակարգի զարգացման հիմնական ասպեկտներ են: Այս հոդվածում առաջարկվում է Unity շարժիչի հեռավորության հաշվարկման ալգորիթմի լավարկում: Բացի այդ, դիտարկվում են Unity շարժիչի կողմից տրամադրվող տվյալների պահպանման մեխանիզմները, և որպես նպատակահարմար տարբերակ, ընտրվում է ֆայլային համակարգը: Իրականացվում է օբյեկտի կոորդինատների պահպանումը ֆայլային համակարգում: Ստացված արդյունքները հիմք են հանդիսանում մասնագիտական հետազոտությունների վիրտուալ ստենդների ստեղծման համակարգի մշակման համար:

Բանալի բառեր՝ Վիրտուալ իրականություն, տվյալների կառավարում, ֆայլային համակարգ, սերիալիզացիա

Обработка и сохранение данных в системах виртуальной реальности

Арман А. Оганесян

Национальный политехнический университет Армении
e-mail: aahovhannisyan1@gmail.com

Аннотация

Обработка и сохранение данных являются ключевыми аспектами разработки системы виртуальной реальности. В данной статье предлагается улучшение алгоритма расчета расстояний Unity Engine. Кроме того, рассматриваются механизмы сохранения данных, предоставляемые Unity Engine, и в качестве подходящего варианта выбирается файловая система. Реализуется хранение координат объекта в файловой системе. Результаты обеспечивают основу для разработки системы создания виртуальных стендов для профессиональных исследований.

Ключевые слова: Виртуальная реальность, управление данными, файловая система, сериализация

Mathematical Problems of Computer Science 58, 91–98, 2022.
doi: 10.51408/1963-0096

UDC 004.7

Network Management Automation Through Virtualization

Arusyak D. Manasyan

Institute for Informatics and Automation Problems of NAS RA
e-mail:armanasyan@iiap.sci.am

Abstract

The study aims to develop methods for automating network management by analyzing its virtual counterpart. The paper substantiates the relevance of this approach, identifies the advantages and disadvantages, highlights the existing problems, and suggests ways to solve them. As a result, the effectiveness of network virtualization was shown by the example of an experimental network.
Keywords: network, automation, virtualization, SDN (Software-Defined Network), OpenDaylight (Software), OpenFlow (Protocol).
Article info: Received 10 December 2021; received in revised form 8 July 2022; accepted 23 August 2022.

1. Introduction

The more devices are connected to the network, the more inconvenience there will be with the expenses of their utilization. And until the network system is automated, this problem will be constant. Organizations will spend a lot of money to buy powerful network devices, but network management will not become easier. That is why a study of network automation and virtualization was carried out, their current applications were discussed and solutions to existing problems were proposed.

As network traffic continues to grow, companies increasingly require large-scale network configurations. The move to cloud computing continues as enterprise customers and their applications rely more and more on network efficiency, so networks are expected to be highly reliable with minimal downtime. As the number of devices on the network increases, so does the need for uninterrupted, flexible, fast, and efficient communication between them. To do this, it is necessary to obtain a large number of network devices that will be of a high quality, and have great features, such as a large amount of memory, many interfaces, and powerful processors, and all this is associated with high costs, which is one of the main prerequisites for the emergence automation and virtualization concepts.

For service providers, automation is a key strategy to improve network agility and reliability while controlling operating and capital costs. Therefore, it is necessary to automate the work with network equipment. Automation of daily network tasks and functions, as well as automated monitoring of iterative processes, increases the availability of network services.

We can describe the current state of the networking industry as "critical". The market-dominant closed (proprietary) solutions are "boxes" for applications, and the interoperability of solutions from different vendors is best provided at the interface level. Networks are extremely complex, making them difficult to scale, manage, and trust. This slows down the further development of networks and programs running in them. Therefore, several solutions for network automation have been developed, and we talked about SDN in our research work. Software Defined Networking (SDN) introduces network virtualization capabilities, which makes it easier to build and manage network automation tasks. Using SDN, networks can be provisioned at the software layer, abstracting the underlying physical hardware. This takes automation to the next level and significantly accelerates network provisioning and configuration management. It also enables IT to attach network and security services to workloads using a policy-driven approach(see [1]). Today, network automation solutions allow us to perform a wide range of tasks, including network planning - design, including scenario planning - backup management, device testing - configuration testing, deployment of deployed physical devices - services, as well as virtual device deployment - provisioning devices, real-time network data collection systems related to applications, network topology, traffic, services, data analysis, including active artificial intelligence, machine learning analysis, to get an idea of the present and future, network behavior, check configuration compliance, to ensure all network devices and service requirements, software updates, including backing up software if necessary, fixing closed network issues, including troubleshooting, and complex, difficult-to-detect Troubleshooting activities, detailed analysis of reports, panels, alarms, warnings, compliance with security requirements, monitoring of the network and its services, service level to maintain customer satisfaction.

The purpose of this article is to show the benefits of network virtualization, present the tools necessary for this, and show its effectiveness as a result of the experimental application.

2. Analyses and Discussion

Network automation through SDN (see [2]) adds a number of capabilities to conventional automation paradigms, which optimize IT resources and require SDN as a networking architecture approach. It enables the control and management of networks using software applications. Through SDN networking, the behavior of the entire network and its devices is programmed in a centrally controlled manner through software applications using open APIs. SDN improves performance through network virtualization. In SDN[2] software-controlled applications or APIs work as a basis of complete network management that may be directing traffic on a network or communicating with underlying hardware infrastructure. So to put it simply, we can say that SDN can create virtual networks or control traditional networks with the help of software to improve security and reduce cost.

Traditional network refers to the old conventional way of networking, which uses fixed and dedicated hardware devices such as routers and switches to control network traffic. Inability to scale, as well as network security and performance are major concerns nowadays in the current growing business situation so SDN is taking control of traditional networks. The traditional network is static and based on hardware network appliances. Traditional network architecture was used by many companies until recent years but nowadays due to its drawbacks SDN has been developed and will be used more widely in the coming years(see [3]).

Table 1. Comparison of SDN to Traditional Network(see [3]).

No	SDN	Traditional Network
1	Virtual networking approach.	Old conventional networking approach.
2	Centralized control.	Distributed control.
3	Programmable network.	This network is nonprogrammable.
4	Open interface.	Closed interface.
5	Data plane and control plane are decoupled by software.	Data plane and control plane are mounted on the same plane.
6	It supports automatic configuration so it takes less time.	It supports static/manual configuration so it takes more time.
7	It can prioritize and block specific network packets.	It leads all packets in the same way with no prioritization support.
8	It is easy to program as per need.	It is difficult to program again and replace the existing program as peruse.
9	The cost is low.	The cost is high.
10	Structural complexity is low.	Structural complexity is high.
11	Extensibility is high.	Extensibility is low.
12	It is easy to troubleshoot and report as it is centralized and controlled.	It is difficult to troubleshoot and report as it is distributed and controlled.
13	Its maintenance cost is lower than the traditional network.	Cost is higher than SDN.

As the SDN technology (see [4]) is based on an intelligent controller, it allows you to automatically redistribute traffic. It turned out that the device allows you to centrally change the settings of network equipment in branches, monitor the network status, load and quality of channels online, and solve problems. This ensures the transparency of data transmission networks and reduces the burden on IT professionals serving the network.

The study also showed that the SDN solution involves the automatic networking of private networks and the transmission of information through all available channels without losing the speed and quality of applications. For example, in the past, only expensive VPN channels were used to transmit audio or video without distortion. Now, thanks to SDN, we can only use the Internet and LTE as a backup(see [5]). In this way, customers can save on telecommunication bill payments and solve VPN reservation issues simply and cheaply. Unlike other virtualization technologies, the open-source SDN solution is more promising. SDN[2] already provides companies with many options to choose from OpenFlow, NETCONF, OVSDB, switches that support the API library, as well as enterprise software that utilizes these protocols. Like any other infrastructure, the SDN infrastructure is built on open standards. This open ecosystem accelerates network innovation. Although the traditional approach to building a network infrastructure still prevails due to the negative impact of mental inertia and crisis events, SDN already allows you to effectively solve problems in a virtual physical environment.

By automating the network, we get the following benefits and services: reduced problems, reduced costs, increased network flexibility, reduced network outages, increased number of strategic employees, advanced analysis, and network management capabilities.

3. Methods and Applications

The article methodology includes the study of epistemological issues, programs (OpenDaylight), protocols (OpenFlow) in the field of networks, using scientific literature, and research articles.

The research aim is to present an example of an automated network as a result of the analysis based on the studied materials. Below is the physical experimental network represented by the GNS3 simulator, which is fully operational, we will get the virtualized version of the following network, but the initial settings must be done one way or another.

This article provides a brief overview of virtual networks and network performance evidence. The physical network shown below is represented by a fully running GNS3 simulator. It contains hosts, routers (Mikrotik), and a virtual switch - OpenvSwitch.

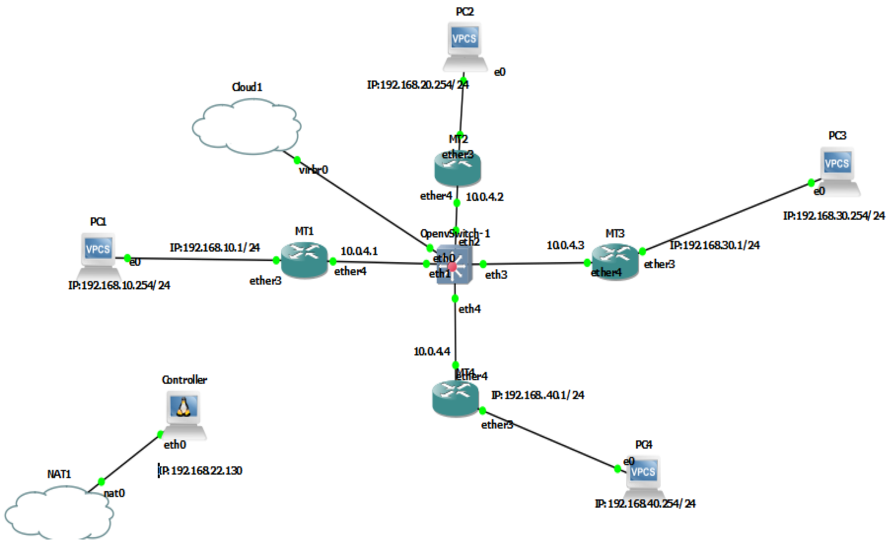


Fig . 1 . Network presented with GNS3 simulator.

Here are the settings of one of the devices, almost the same as the rest:

```
/routing OSPF instance
set [ find default=yes ] router-id=10.255.255.1
/IP address
add address=10.0.4.1/24 interface=ether4 network=10.0.4.0
add address=192.168.10.1/24 interface=ether3 network=192.168.10.0
/routing OSPF network
add area=backbone network=10.0.4.0/24
add area=backbone network=192.168.10.0/24
```

Here are the minimum settings that make the network complete.
For network virtualization, as mentioned at the beginning, we implemented an SDN solution. We have demonstrated the use of SDN with the OpenDaylight software, which is a software platform for SDN.

To work with our controller, to connect it to our physical network, we downloaded and activated the following components:

```
opendaylight-user@root>feature:install odl-restconf odl-l2switch-all odl-mdsal-apidocs
odl-dlux-all odl-openflowplugin-all
```

They provide a graphical user interface of OpenDaylight software, as well as the necessary tools and devices. After activating them, immediately after setting the appropriate settings in our physical OpenvSwitch network, we see a virtualized version of our network.

To establish a "controller" connection in our physical network, we have previously configured the OpenvSwitch OpenFlow device by giving it the IP address of the controller by typing the following command: **ovs-vsctl set-controller br0 tcp: 192.168.18.129:6633**, where 192.168.18.129 is the IP address of the controller and it can be different for different devices, 6633 is the connection port and the protocol that controls data transfer over TCP. Thanks to this, it was able to communicate with other devices.

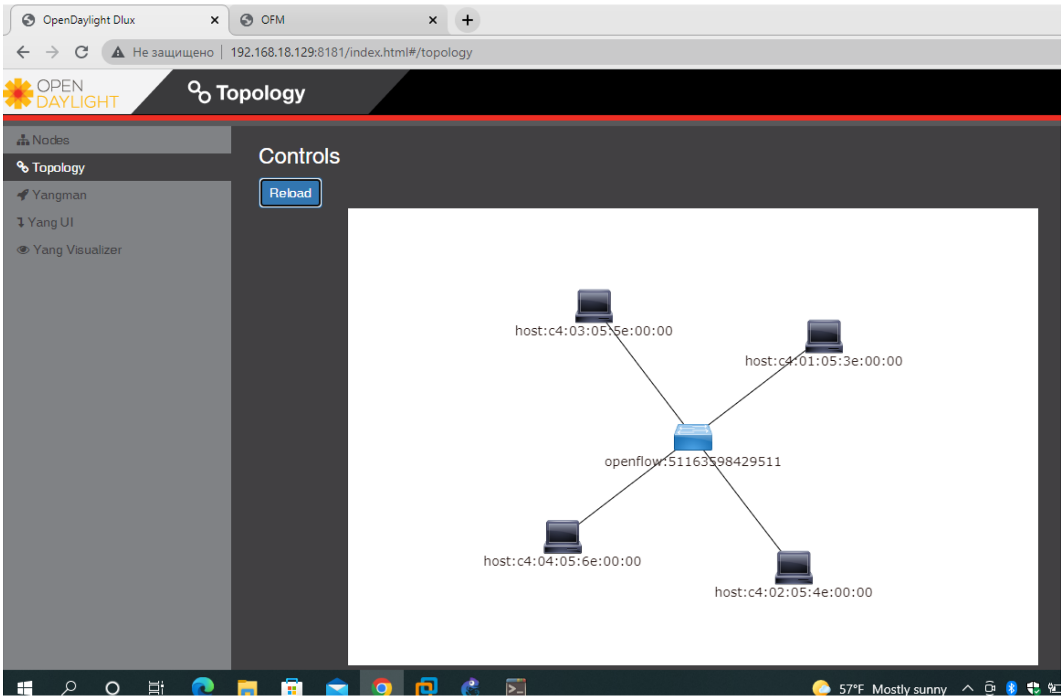


Fig. 2 . Example of a virtual network in OpenDaylight.

Fig. 2 shows a virtualized version of the physical network in OpenDaylight. The picture clearly shows all the devices in our network that are connected to the OpenFlow protocol support device, OpenvSwitch. It is thanks to the OpenFlow protocol that our SDN controller sees our entire physical network.

OpenFlow is a protocol for managing data processing, which is transmitted over the network through routers and switches using SDN technology. Fast packet forwarding (data forwarding) on a classic router or switch and high-level routing decisions (control operations) are made on the same device. The OpenFlow switch separates these two functions. Data redirection is performed by the switch itself, while routing decisions are entrusted to a separate controller, usually a standard server.

After clicking on the network topology, Yang automatically shows us the CONFREST API URL it uses to get this information:

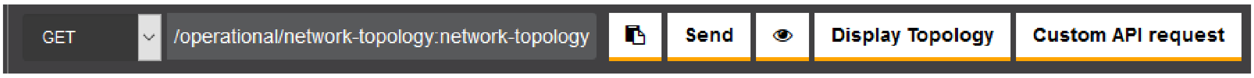


Fig. 3. CONFREST API URL

By clicking the send button(Fig.3), we can see the topology of our operational network.

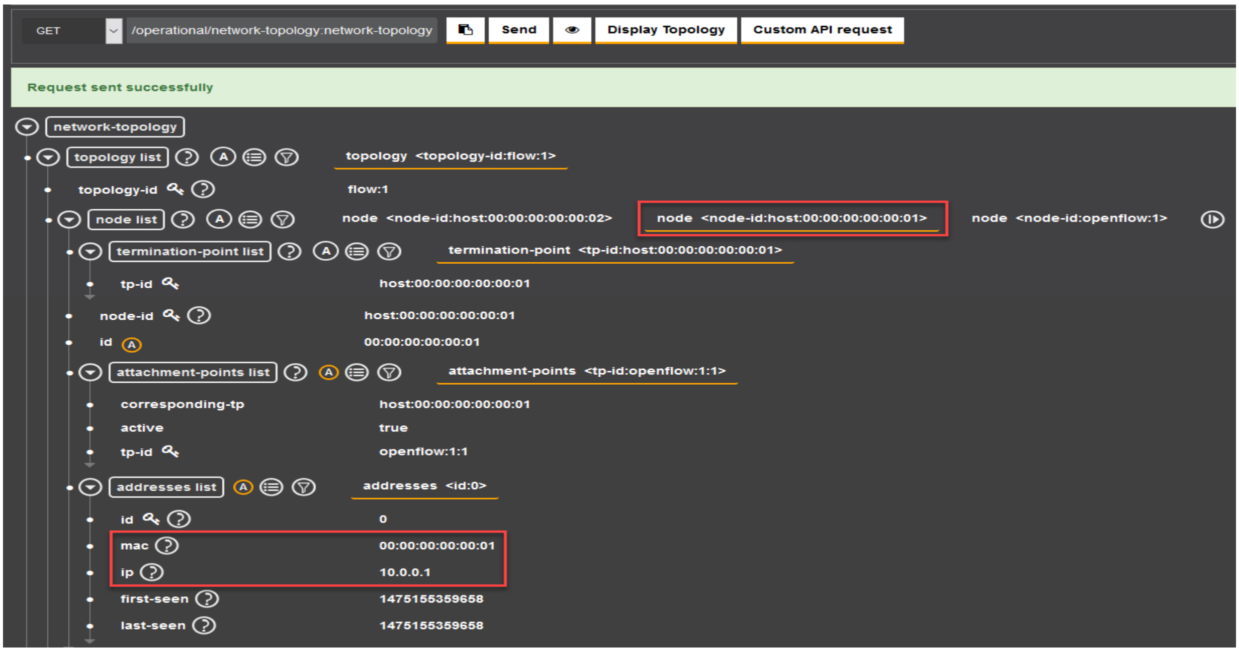
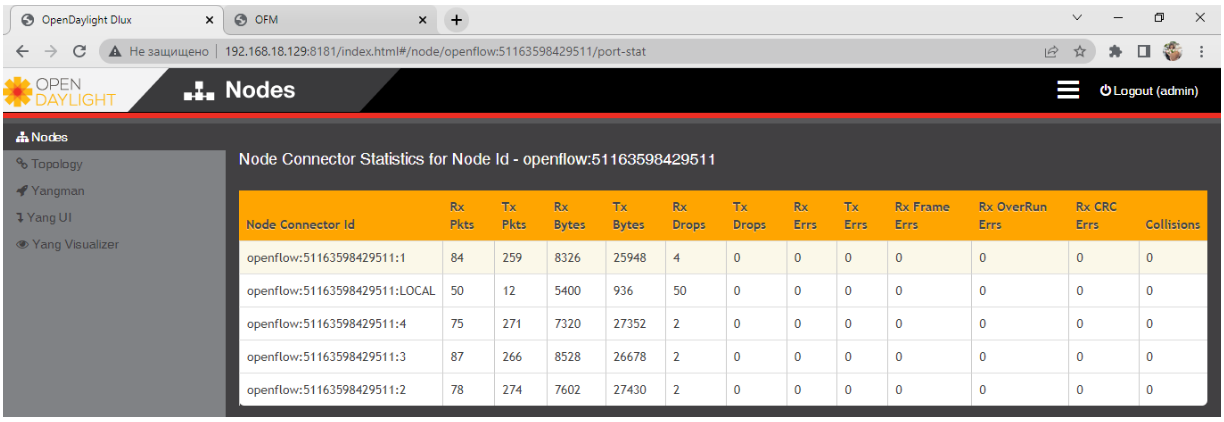


Fig. 4. Operating network topology.

In Fig. 4, we can see information about our current topology, including the MAC (Media Access Control) and IP addresses of our hosts. So, you do not need to enter the device to see them every time, but you can see them from one control panel of SDN.

When we send network traffic, all the information about it is mentioned in the flow tables of the SDN: how many packages were sent to us, how many arrived, how many dropped on the way, and what errors we encountered. And all that information we can see in the nodes of Fig.5.



Node Connector Id	Rx Pkts	Tx Pkts	Rx Bytes	Tx Bytes	Rx Drops	Tx Drops	Rx Errs	Tx Errs	Rx Frame Errs	Rx OverRun Errs	Rx CRC Errs	Collisions
openflow:51163598429511:1	84	259	8326	25948	4	0	0	0	0	0	0	0
openflow:51163598429511:LOCAL	50	12	5400	936	50	0	0	0	0	0	0	0
openflow:51163598429511:4	75	271	7320	27352	2	0	0	0	0	0	0	0
openflow:51163598429511:3	87	266	8528	26678	2	0	0	0	0	0	0	0
openflow:51163598429511:2	78	274	7602	27430	2	0	0	0	0	0	0	0

Fig.5. Node connector statistics.

5. Conclusion

This paper proposes a solution for network optimization. As a result of the research, we concluded that automation improves the speed of IT operations in response to analytical change. The ability to monitor operations, just as needed, provides greater visual control of the network, and transparency of processes within it. Network automation improves work efficiency, reduces human error, increases access to network services, and provides better customer service. Research has shown that the SDN solution includes the automatic integration of private networks, and the transmission of information over all available channels, without loss of application speed and quality. As a result of the study, it became clear that network automation can be implemented regardless of its type, which facilitates its transition. Network virtualization is a more all-encompassing version of virtualization that makes it possible to convert physical network hardware into software that can easily be transitioned to different domains as needed, increasing flexibility and scalability for the network. I came to the conclusion that its use on the network will be of great benefit to network administrators.

References

[1] How Can Network Automation Be Improved With SDN. [Online]. Available: https://www.acadiatech.com/blog/how-can-network-automation-improve-with-sdn/

[1] W. Braun and M. Menth, "Software-defined networking using OpenFlow: Protocols applications and architectural design choices", *Future Internet*, vol. 6, no. 2, pp. 302-336, 2014, [Online]. Available: http://www.mdpi.com/1999-5903/6/2/302

[2] S. Jena. Difference between Software Defined Network and Traditional Network. [Online]. Available: https://www.geeksforgeeks.org/difference-between-software-defined-network-and-traditional-network/

[3] J. Doherty, SDN and NFV Simplified: A Visual Guide to Understanding Software Defined Networks and Network Function Virtualization, Addison-Wesley Professional, March 2016.

[4] Software-Defined SD-WAN Technologies. [Online]. Available: https://tadviser.com/index.php/Article:SD-WAN_%28Software_Defined%29_Software-Defined_WAN

Ցանցի ղեկավարման ավտոմատացում վիրտուալացման միջոցով

Արուսյակ Դ. Մանասյան

ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ
e-mail:armanasyan@iiap.sci.am

Ամփոփում

Հետազոտության նպատակն էր մշակել ցանցի ղեկավարման ավտոմատացման մեթոդներ՝ վերլուծելով դրանց վիրտուալ անալոգը: Աշխատանքում հիմնավորվում է այս մոտեցման արդիականությունը, վեր են հանվում առավելություններն ու թերությունները, ընդգծվում են առկա խնդիրները և առաջարկվում են դրանց լուծման ուղիներ: Արդյունքում ներկայացվել է ցանցերի վիրտուալացման արդյունավետությունը՝ փորձնական ցանցի օրինակով:

Բանալի բառեր՝ ցանց, ավտոմատացում, վիրտուալացում, SDN (Ծրագրակողմնորոշված ցանց), OpenDaylight (Ծրագրային ապահովում), OpenFlow (Արձանագրություն):

Автоматизация управления сетью за счет виртуализации

Арусяк Д. Манасян

Институт проблем информатики и автоматизации НАН РА
e-mail:armanasyan@iiap.sci.am

Аннотация

Цель исследования заключалась в разработке методов автоматизации управления сетью путем анализа ее виртуального аналога. В работе обосновывается актуальность такого подхода, выявляются преимущества и недостатки, подчеркиваются существующие проблемы и предлагаются пути их решения. В результате была показана эффективность виртуализации сети на примере пилотной сети.

Ключевые слова: сеть, автоматизация, виртуализация, SDN (Программно-определяемая сеть), OpenDaylight (Программное обеспечение), OpenFlow (Протокол).

Կանոններ հեղինակների համար

ՀՀ ԳԱԱ ԻԱՊԻ "Կոմպյուտերային գիտության մաթեմատիկական խնդիրներ" պարբերականը տպագրվում է 1963 թվականից: Պարբերականում հրատարակվում են նշված ոլորտին առնչվող գիտական հոդվածներ, որոնք պարունակում են նոր` չհրատարակված արդյունքներ:

Հոդվածները ներկայացվում են անգլերեն` ձևավորված համապատասխան "ոճով" (style): Հոդվածի ձևավորման պահանջներին ավելի մանրամասն կարելի է ծանոթանալ պարբերականի կայքէջում` http://mpcs.sci.am/:

Rules for authors

The periodical "Mathematical Problems of Computer Science" of IIAP NAS RA has been published since 1963. Scientific articles related to the noted fields with novel and previously unpublished results are published in the periodical.

Papers should be submitted in English and prepared in the appropriate style. For more information, please visit the periodical's website at http://mpcs.sci.am/.

Правила для авторов

Журнал «Математические проблемы компьютерных наук» ИПИА НАН РА издается с 1963 года. В журнале публикуются научные статьи в указанной области, содержащие новые и ранее не опубликованные результаты.

Статьи представляются на английском языке и оформляются в соответствующем стиле. Дополнительную информацию можно получить на веб-сайте журнала: http://mpcs.sci.am/.

The electronic version of the periodical “Mathematical Problems of Computer Science” and rules for authors are available at

http://mpcs.sci.am/

Phone:	(+37460) 62-35-51
Fax:	(+37410) 28-20-50
E-mail:	mpcs@sci.am
Website:	http://mpcs.sci.am/

Ստորագրված է տպագրության՝ 25.11.2022
Թուղթը՝ օֆսեթ:
Հրատարակված է ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտի կողմից
Ծավալը՝ 100 էջ: Տպաքանակը՝ 100
ՀՀ ԳԱԱ ԻԱՊԻ Համակարգչային պոլիգրաֆիայի լաբորատորիա
Երևան, Պ. Սևակի 1
Հեռ. +(374 60) 623553
Գինը՝ անվճար

Подписано в печать 25.11.2022
Офсетная бумага.
Опубликовано Институтом проблем информатики и автоматизации НАН РА
Объём: 100 страниц. Тираж: 100
Лаборатория компьютерной полиграфии ИПИА НАН РА.
Ереван, П. Севака 1
Тел.: +(374 60) 623553
Цена: бесплатно

Signed in print 25.11.2022
Offset paper
Published by Institute for Informatics and Automation Problems of NAS RA
Volume: 100 pages
Circulation: 100
Computer Printing Lab of IIAP NAS RA
Yerevan, 1, P. Sevak str.
Phone: +(374 60) 623553
Free of charge