UDK 004.732.056 DOI: 10.20998/2411-0558.2018.24.07

S. Yu. GAVRYLENKO, PhD Tech., Associate Professor, National technical university "Kharkiv Polytechnic Institute",
O. S. BABENKO, Post graduate student, National technical university "Kharkiv Polytechnic Institute"

DEVELOPMENT AN ANTIVIRUS SCANNER BASED ON THE NEURAL NETWORK ART-1

In this article, the methods for constructing antivirus programs, their advantages and disadvantages are considered. The PE-structure of malicious and safe software was analyzed. The signs that are typical for classes of malware such as Worm, Backdoor, Trojan and for the safe software were identified. A software model of the device for detecting malicious software based on the neural network ART-1 was developed. This system was trained on the example of the obtained binary vectors. Optimal similarity coefficients were found, and the testing was performed. The test results showed the possibility of using the developed system to detect modified malicious software. Figs.: 3. Tabl.: 4. Refs.: 10 titles.

Keywords: antivirus scanner; malicious; PE-structure; software; neural network; ART-1; coefficients.

Formulation of the problem. It’s known that over a year, viruses cause damage to hundreds of billions of dollars, and about the same amount is indirect damage associated with the development of software and other measures to protect against viruses.

The most notable event in 2017 was the virus Petya epidemic. It is the first time in the history of Ukraine, that sites of government and private agencies were struck in a few hours due to hacker’s attack. This malware infected computers of many organizations and individuals in 60 countries around the world. The damage from the virus attack is estimated at $ 8 billion

It becomes obvious that the analysis of such a number of malicious code and the formation of the entire spectrum of virus signature is an almost unrealized task. That is why the actual topic is the development of effective methods and technologies for counteracting computer viruses based on heuristic methods.

Analysis of literature [1 – 4], as well as research on the methods of heuristic analysis in antivirus programs showed a high variety of existing approaches and methods of heuristic analysis: intelligent subsystems based on the theory of artificial intelligence, methods of fuzzy logic, cluster analysis, the theory of neural networks, genetic algorithms and other. The main disadvantage of the heuristic method is the high frequency of false positives.

Statistical methods based on control cards (Shukhart control maps, CUSUM maps, EWMA cards, etc.) can also be used to solve the set tasks [5]. In addition, methods for statistical data processing, for example, BDS testing [6], can be used for further refine of obtained results.

These methods are based on the assumption that for a computer system (CS) there is a template for normal behavior and any significant deviations from it may be due to the influence of intruders. That is why the very important task is to select or form a template, which would reproduce the functional portrait of the CS and record its abnormal behavior with the given accuracy. At the same time, the more input data is analyzed, the more accurate the result of the evaluation. Meanwhile, if the model or evaluation criterion is chosen incorrectly, parametric methods lose their basic authority, which can lead to an increase in false positives.

The conducted studies have shown that the main way of eliminating these shortcomings is to improve the models of information technology and the reasoned choice of criteria for evaluating abnormal behavior of computer systems.

On the one hand neural networks, can behave as a deterministic machine, on the other as a fuzzy system, evaluating new data that did not participate in the formation of the neural network. Such a result is achieved by learning networks [8, 9] and not by the formation of reaction rules, as is done in classical approaches. Ability of the networks to educate by examples makes them more attractive in comparison to systems that function according to a defined system of rules formulated by experts. The education process can be considered the architectural determination of the network and finding the coefficients of the connections between neurons. The neural network adjusts the weight of the connections depending on the existing training set and is a parallel computing device, since it is based on a set of simple computing elements - the parallel functioning neurons. Such a system is resistant to damage, that is, the network will work even if part of the neurons fails. In the education process, the neural network is capable to detect complex interdependencies between incoming and output data, as well as generalization. This means that in case of successful training, the network will be able to return the correct result based on the data that was missing in the training sample, as well as incomplete or partially distorted data.

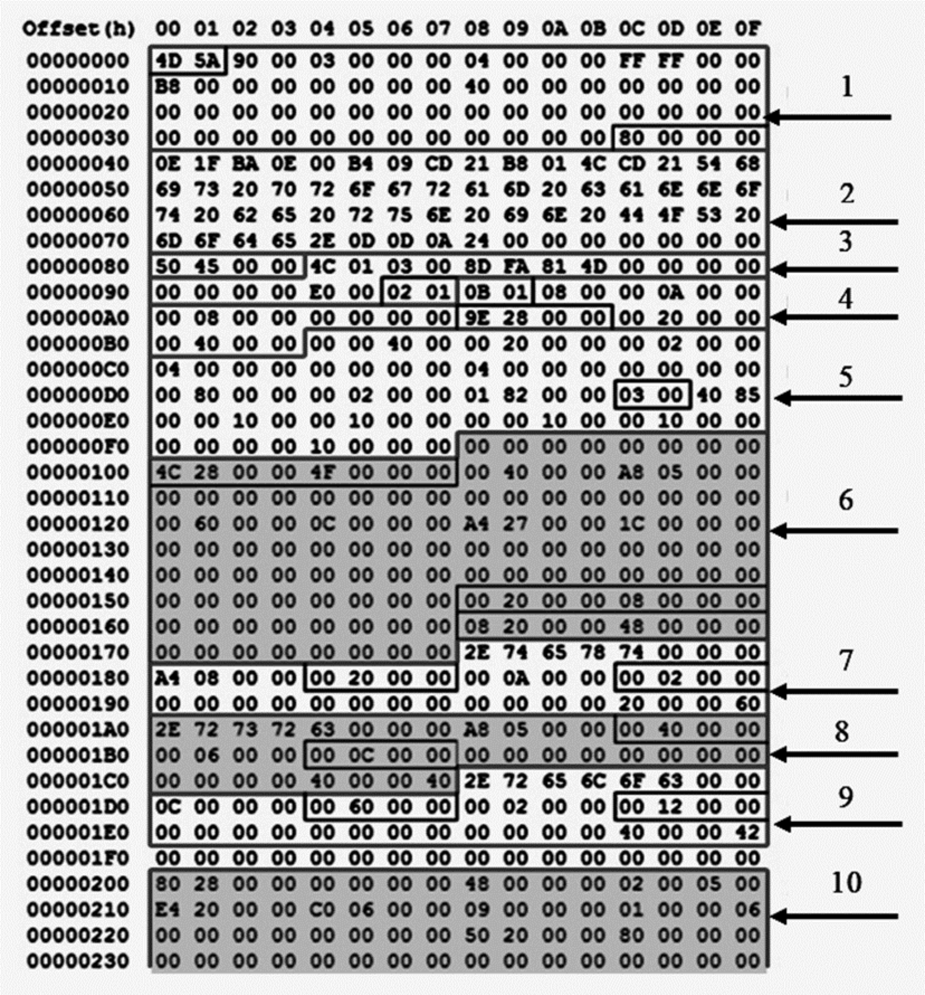
An essential disadvantage of most neural networks is the inability to learn at the appearance of new information (lack of stability-plasticity). Therefore, to develop a heuristic analyzer, it was decided to use one of the few types of neural networks that possesses this property: a discrete neural network of adaptive resonance (ART-1). The algorithm for training the neural network is given in [10].

The **purpose of this article** is to develop a system for detecting computer viruses based on the ART-1 neural network.

Results of the development and research. The conducted researches have shown that one of the perspective directions of heuristic analysis of computer viruses is the use of neural networks [7-10].

The input data for training the neural network are based on the analysis of the PE structure of the file.

In the Fig.1 it is showed an example of PE structure of the file and emphasized on the areas for the further analysis.



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00
00000010  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000030  00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 00
00000040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68
00000050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F
00000060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20
00000070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00
00000080  50 45 00 00 4C 01 03 00 8D FA 81 4D 00 00 00 00
00000090  00 00 00 00 E0 00 02 01 0B 01 08 00 00 0A 00 00
000000A0  00 08 00 00 00 00 00 00 9E 28 00 00 00 20 00 00
000000B0  00 40 00 00 00 00 40 00 00 20 00 00 00 02 00 00
000000C0  04 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00
000000D0  00 80 00 00 00 02 00 00 01 82 00 00 03 00 40 85
000000E0  00 00 10 00 00 10 00 00 00 00 10 00 00 10 00 00
000000F0  00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00
00000100  4C 28 00 00 4F 00 00 00 00 40 00 00 A8 05 00 00
00000110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000120  00 60 00 00 0C 00 00 00 A4 27 00 00 1C 00 00 00
00000130  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000150  00 00 00 00 00 00 00 00 00 20 00 00 08 00 00 00
00000160  00 00 00 00 00 00 00 00 08 20 00 00 48 00 00 00
00000170  00 00 00 00 00 00 00 00 2E 74 65 78 74 00 00 00
00000180  A4 08 00 00 00 20 00 00 00 0A 00 00 00 02 00 00
00000190  00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 60
000001A0  2E 72 73 72 63 00 00 00 A8 05 00 00 00 40 00 00
000001B0  00 06 00 00 00 0C 00 00 00 00 00 00 00 00 00 00
000001C0  00 00 00 00 40 00 00 40 2E 72 65 6C 6F 63 00 00
000001D0  0C 00 00 00 00 60 00 00 00 02 00 00 00 12 00 00
000001E0  00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 42
000001F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000200  80 28 00 00 00 00 00 00 48 00 00 00 02 00 05 00
00000210  E4 20 00 00 C0 06 00 00 09 00 00 00 01 00 00 06
00000220  00 00 00 00 00 00 00 00 50 20 00 00 80 00 00 00
00000230  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Fig 1. An example of PE file structure,

where: 1 – DOS header section; 2 – DOS message section; 3 – PE header; 4 – standard PE fields; 5 – PE NT fields; 6 – Catalogue data; 7 – .text header section; 8 – .rscs header section; 9 – .reloc header section; 10 – .text section.

The PE structure of harmful and secure software was analyzed, namely:

– 290 Worm type files;

– 1050 files such as Trojan;

– 1153 files of type Backdoor;

– 1000 secure files.

As a result of the analysis of the PE structure of the investigated files received:

– a table with API-functions and libraries in which they are included. There were 24945 records received.

– a table with strings. Found in total 175651 rows, their length varied from 6 to 70 characters.

The analysis of the data received from malicious and secure softwareallowed to highlight functions and rows, and to form the signs (tabl. 1), which are inherent to the considered viruses and to form a table of attributes. For further analysis it was decided to use 49 signs.

These signs were later used as bitmaps for file analysis. The binaries of malicious files like Worm, Backdoor, Trojan and secure software were obtained as a result of searching for selected attributes in files.

For the correct functioning of the state identification system, an optimal coefficient of similarity has been selected experimentally. The optimal coefficient is the coefficient at which there are no false-positives on the learning sample.

At the first stage, 100 signatures of the type Backdoor are taken for basic knowledge of the neural network. The initial similarity factor is intended to be 0.6, since it is the minimum allowable for the neural network. Signatures of 100 secure files were submitted to the input of the neural network for recognition. As a result of the program's operation, some of the input signatures were displayed on the console. Console-derived signatures were attributed to signatures of the type Backdoor, which was a mistake (fig. 2) and required an increase of the coefficient of similarity to the optimal. The optimal coefficient is considered to be the coefficient for which there are no false recognitions. The results of the experimental selection of the optimal similarity coefficient for malicious software of the type Backdoor are given in tabl. 2. Thus, the absence of false positives for malicious software such as Backdoor managed to achieve with a similarity factor of 0.84.

At the second and third stage, 100 signatures of malicious software such as Trojan and Worm are taken for basic knowledge of the neural network. The results of the experimental selection of the optimal coefficient of similarity are given in Table 3 and Table 4.

Table 1

Features malicious signatures

	Functions/lines	Var 1	Var 2	Quantity in the viruses
1	callnexthookex	-	-	42
2	copyfile	CopyFileA	CopyFileW	132
3	createfile	CreateFileA	CreateFileW	178
4	enumcalendarinfo	EnumCalendarInfoA	-	43
5	findfirstfile	FindFirstFileA	FindFirstFileW	127
6	getcurrent processid	-	-	70
....				
37	host	-	-	10
38	admin	-	-	-
39	hostname	-	-	9
40	localhost	-	-	19
41	sendmail	-	-	28
42	userhost	-	-	9
43	wininet	-	-	35
44	wnet	WNetEnumResourceA	WNetOpenEnumA	55
45	getmodulefile name	GetModuleFileNameA	GetModule FileNameW	206
46	getmodulehandle	GetModuleHandleA	-	247
47	loadlibrary	LoadLibraryA	-	135
48	setfilepointer	SetFilePointerEx	-	158
49	virtualalloc	VirtualAllocEx	-	158

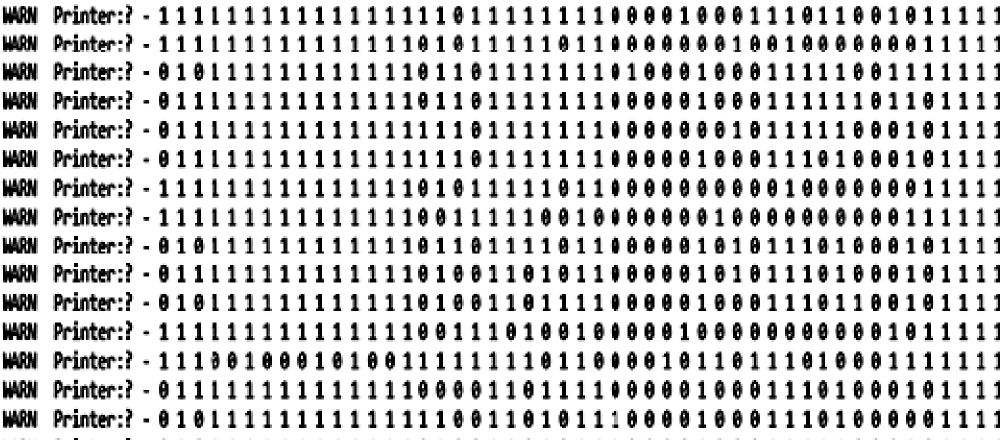
Table 2

Experimental match of the optimal coefficient of similarity for the viruses type Backdoor

Coefficient of similarity	Quantity of false alerts for the safe signatures
0,60	15
0,70	10
0,80	1
0,84	0

At the initial base of Trojan malware, Trojan – 100%, Backdoor – 46%, Worm – 1% were detected.

The results of the identification system showed that when training the neural system by the Backdoor sample, the system also begins to identify harmful Trojan (28%) signatures, since these types of signatures have a high similarity coefficient, since they perform similar actions from the point of view of the operating system.



```
WARN  Printer:? - 1111111111111111101111111111100001000111011001011111
WARN  Printer:? - 1111111111111110101111101101100000001001000000011111
WARN  Printer:? - 0101111111111110110111111111101000100011111001111111
WARN  Printer:? - 0111111111111110110111111111100000100011111101101111
WARN  Printer:? - 0111111111111111110111111111100000001011111000101111
WARN  Printer:? - 0111111111111111110111111111100000100011101000101111
WARN  Printer:? - 1111111111111110101111101101100000000001000000011111
WARN  Printer:? - 1111111111111110011111001001000000010000000000011111
WARN  Printer:? - 0101111111111110110111101101100000101011101000101111
WARN  Printer:? - 0111111111111110100110101101100000101011101000101111
WARN  Printer:? - 0101111111111110100110111111100000100011101100101111
WARN  Printer:? - 1111111111111110011101001001000001000000000000101111
WARN  Printer:? - 1110010001010011111111011011000010110111010001111111
WARN  Printer:? - 0111111111111110000110111111100000100011101000101111
WARN  Printer:? - 0101111111111111100110101101110000100011101000001111
```

Fig. 2. Mistakes making by coefficient 0,6

In the future, 3 experiments with different input data and different similarity coefficients were carried out. The first one uses 100 signatures of the type Backdoor as the base of initial knowledge, 400 signatures are presented to the recognition input, which represent a mixture of all possible signatures (100 Backdoor signatures, 100 Trojan signatures, 100 Worm signatures, 100 signature files of secure files). The second one is the base of 100 signatures of type Trojan, the input is 400 signatures. The third is the base of 100 signatures of type Worm, the input is 400 signatures.

At the initial base of Backdoor malicious signatures, an additional group of malicious Trojan 28% signature was detected

At the initial base of Warm malicious signatures, only signatures from this group were 100%

Table 3

Experimental match of the optimal coefficient of similarity for the viruses type Trojan

Coefficient of similarity	Quantity of false alerts for the safe signatures
0,6	10
0,65	3
0,7	0

When learning a worm-patterned system, only a signature of this type is detected, due to a relatively high optimal similarity factor of 0.97, the tokens of this type are abandoned from Signature Backdoor, Trojan, and secure software.

Table 4

Experimental match of the optimal coefficient of similarity for the viruses type Worm

Coefficient of similarity	Quantity of false alerts for the safe signatures
0,6	29
0,7	14
0,8	17
0,9	10
0,97	0

Also, when training a neural system with Trojan sampling, in addition to this sample, the system recognizes backdoor signatures (46%) and insignificant number of worm-signatures (1%).

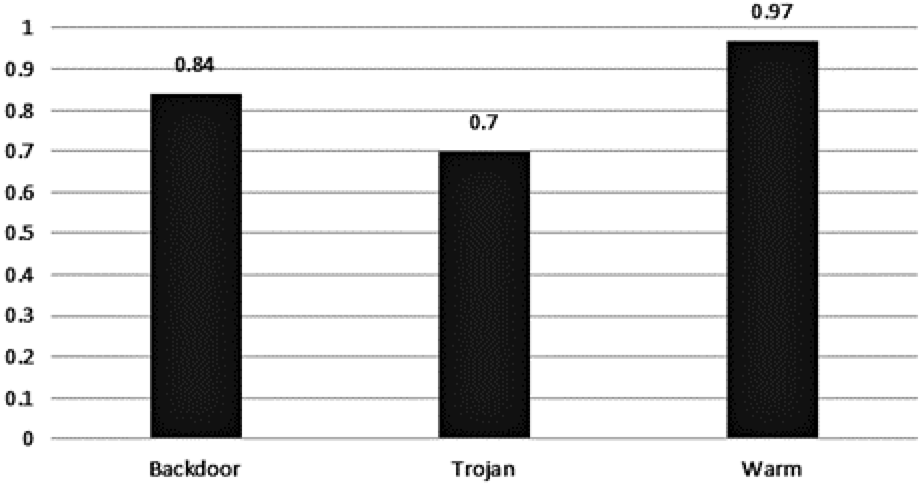


Fig. 3. Optimal coefficients of the similarity

This percentage of signature recognition (Backdoor - Trojan and vice versa) is due to the fact that these types of viruses have similar actions from the point of view of the operating system, namely harm to it: the desire to obtain unauthorized access to data or remote control of the operating system and the computer as a whole; collection of information and its transmission to the attacker, its destruction or malicious change, the disruption of the computer; the use of computer resources for other purposes.

Conclusions. In this article, the methods of constructing antivirus programs are analyzed. The RE structure of harmful and secure software has been analyzed, API functions and rows are found, inherent to these files, and selected part of them for further analysis. The result of the search for highlighted features in files is binary vectors of malicious software such as Worm, Backdoor, Trojan and secure software, and used as inputs for training the neural network.

The software model of the heuristic analyzer on the basis of the ART-1 neural network has been developed, the optimal similarity coefficients were found, and the tested computer virus detection system was tested. System was modified in order to use three parallel ART-1 blocks working in parallel, that were studied with different set of data.

The test results showed that when training the neural system with the Backdoor sample, the system also begins to identify 28% of the malicious Trojan type signatures, since these types of signatures perform such actions from the point of view of the operating system. When learning the system, the worm sample is only susceptible to virus of this type.

The obtained research results showed the possibility of using the developed system of identifying malicious software as an auxiliary method in information security systems of the COP.

References

1. Lukatsky, A. (2001), *Attack Detection*, VHV-Petersburg, St. Petersburg, 624 p.

2. Shelukhin, O., Sakalema Zh., and Filinov A. (2013) *Intrusion Detection into Computer Networks,* Hot line-Telecom, Moscow, 220 p.

3. Semenov, S., Davydov, V., and Gavrilenko S. (2014), *Data Protection in Computer-Aided Control Systems*, "Lap Lambert Academic Publishing", Germany, 236 p.

4. Goshko, S. (2009), *Technologies of fight against computer viruses,* Solon Press, Moscow, 352 p.

5. Gavrilenko, S., Chelak, V., and Hornostal, O. (2016), Intrusion detection in computer systems, *Proceedings of the symposium "Metrology and metrology assurance",* Sozopol, Bulgaria, pp. 342-347.

6. Semenov, S., Gavrilenko, S., and Chelak, V. (2016), Development of templates for the identification of the state of computer systems based on BDS-testing, *Herald of the National Technical University "KhPI". Subject issue: Information Science and Modelling*, Vol. 21, pp.118-125.

7. Grishin, A. (2011), Neural network technologies in the tasks of detecting computer attacks, *Information technologies and computing systems*, Vol. 1, pp. 53-64.

8. Bezobrazov, S., and Golovko, V. (2010), Algorithms of artificial immune systems and neural networks for detecting malware, *Neuroinformatics*, Vol. 7, pp. 273-288.

9. Dmitrienko, V., Zakovrotny, A., Noskov, V., and Mezentsev, M. (2014), *Fundamentals of neurocomputing: educational and methodical manual for practical classes*, Kharkov: HTMT, Kharkov, 140 p.

10. Gavrilenko, S., Semenov, S., and Babenko, A. (2015), Development of a computer virus detection system based on the ART-1 neural network, *Systems of information processing*, Vol. 10 (135), pp 126-129.

Статью представил д-р техн. наук, проф. НТУ "ХПИ" Семенов С.Г.

Received 05.05.2018

Gavrilenko Svitlana, PhD Tech.,
National Technical University " Kharkov Polytechnic Institute",
Str. Kirpicheva, 21, Kharkov, Ukraine, 61002,
Tel.:+38-057-707-01-65, e-mail: gavrilenko08@gmail.com,
ORCID ID:0000-0006-4561-8368

Babenko Oleksii, Post-graduate student,
National Technical University " Kharkov Polytechnic Institute",
Str Kirpicheva, 21, Kharkov, Ukraine, 61002,
Tel.: +38-093-301-33-39, e-mail: alex0128.94@mail.ru,
ORCID ID: 0000-0002-7340-3754

УДК 004.732.056

Розробка антивірусного сканера на основі нейронної мережі АРТ-1 / Гавриленко С.Ю., Бабенко О.С. // Вісник НТУ "ХПІ". Серія: Інформатика та моделювання. – Харків: НТУ "ХПІ". – 2018. – № 24 (1300). – С. 70 – 79.

У статті розглядаються методи побудови антивірусних програм, їх переваги та недоліки. Проаналізовано PE-структура файлів шкідливого і безпечного програмного забезпечення. Виділено ознаки, які притаманні для класів шкідливих програм, таких як Worm, Backdoor, Trojan і для безпечного програмного забезпечення. Розроблено програмну модель для виявлення шкідливого програмного забезпечення на основі нейронної мережі АРТ-1. Виконано навчання системи на прикладі отриманих двійкових векторів. Знайдено оптимальні коефіцієнти подібності, виконано тестування. Результати випробувань показали можливість використання розробленої системи для виявлення модифікованого шкідливого програмного забезпечення. Іл.: 3. Табл.: 4. Бібліогр.: 10 назв.

Ключові слова: антивірусний сканер; шкідливе програмне забезпечення; PE-структура файлу; нейронна мережа ART-1; оптимальний коефіцієнт подібності.

УДК 004.732.056

Разработка антивирусного сканера на основе нейронной сети ART-1 / Гавриленко С.Ю., Бабенко А.С. // Вестник НТУ "ХПИ". Серия: Информатика и моделирование. – Харьков: НТУ "ХПИ". – 2018. – № 24 (1300). – С. 70 – 79.

В статье рассматриваются методы построения антивирусных программ, их преимущества и недостатки. Проанализирована PE-структура файлов вредоносного и безопасного программного обеспечения. Выделены признаки, типичные для классов вредоносных программ, таких как Worm, Backdoor, Trojan и для безопасного программного обеспечения. Разработана программная модель для обнаружения вредоносного программного обеспечения на основе нейронной сети АРТ-1. Выполнено обучение системы на примере полученных двоичных векторов. Найдены оптимальные коэффициенты подобия, выполнено тестирование. Результаты испытаний показали возможность использования разработанной системы для обнаружения модифицированного вредоносного программного обеспечения. Ил.: 3. Табл.: 4. Библиогр.: 10 назв.

Ключевые слова: антивирусный сканер; вредоносное программное обеспечение; PE-структура файла; нейронная сеть ART-1; оптимальный коэффициент подобия.

UDK 004.732.056

Development an antivirus scanner based on the neural network ART-1 / Gavrylenko S. Yu., Babenko O. S. // Herald of the National Technical University "KhPI". Subject issue: Information Science and Modelling. – Kharkov: NTU "KhPI". – 2018. – №. 24 (1300). – P. 70 – 79.

In this article, the methods for constructing antivirus programs, their advantages and disadvantages are considered. The file PE-structure of malicious and safe software was analyzed. The signs that are typical for classes of malware such as Worm, Backdoor, Trojan and for the safe software were identified. A software model of the device for detecting malicious software based on the neural network ART-1 was developed. This system was trained on the example of the obtained binary vectors. Optimal similarity coefficients were found, and the testing was performed. The test results showed the possibility of using the developed system to detect modified malicious software. Figs.: 3. Tabl.: 4. Refs.: 10 titles.

Keywords: antivirus scanner; malicious; PE-structure; software; ART-1 neural network; optimal similarity coefficients.