

-  
\_\_\_\_\_  
( )  
\_\_\_\_\_  
( )

\_\_\_\_\_  
( )  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
( )

:  
\_\_\_\_\_  
II , \_\_\_\_\_ -18-2  
\_\_\_\_\_  
( , )

\_\_\_\_\_  
123 - ' \_\_\_\_\_  
\_\_\_\_\_  
( )

\_\_\_\_\_  
\_\_\_\_\_ - \_\_\_\_\_  
\_\_\_\_\_  
( - - )

\_\_\_\_\_  
( )  
:  
\_\_\_\_\_  
( , , )

\_\_\_\_\_  
( ) \_\_\_\_\_ ( , )



5. ( ) \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

6. .1) ( , , , , ) ,

	( , , , , )		

1		31.03.20-13.04.20	
2		14.04.20-21.04.20	
3	,	22.04.20-27.04.20	
4		28.04.20-05.05.20	
5		06.05.20-11.05.20	
6		12.05.20	

30 2020 .

\_\_\_\_\_  
 ( )  
 | \_\_\_\_\_ ( ) \_\_\_\_\_ ( , , , ) \_\_\_\_\_

: 91 ., 13 ., 3 ., 2

., 30 .

,

,

,

.

,

.

.

,

,

.

.

## ABSTRACT

Master's thesis: 91 pages, 13 figures, 3 tables, 2 appendices, 30 sources.

VIRUS, SOFTWARE, MOBILE SOFTWARE, DEEP TRAINING,  
NEURAL NETWORKS

The relevance of this work is due to the fact that the existing algorithms for finding malicious code are imperfect. For the most part, search engines determine the presence of malicious code based on already known samples.

The purpose of the work is to improve the quality of malware detection by developing an algorithm for signature extraction using deep learning algorithms.

The object of the work is software that runs with elements of malicious code.

The subject of the work is the development of an algorithm for distinguishing features of malicious software and further creating a compact representation of this behavior.

	,	,	,	
	.....			8
	.....			9
1				
	.....			11
1.1		.....		11
1.2	.		.....	13
1.3	.		.....	15
1.3.1		.....		15
1.3.2		.....		16
1.3.3			.....	17
1.3.4			.....	20
1.3.5			.....	21
2			,	
	.....			25
2.1		.....		25
2.2			.....	27
2.3		.....		30
2.3.1			.....	30
2.3.2			.....	32
2.4				
	.....			33
3				
	.....			37
3.1				
	.....			37

3.2	.....	42
3.3	.....	45
3.4	.....	47
4		
	.....	50
4.1	.....	50
4.2	.....	52
4.3	.....	55
4.4	.....	61
4.7		
	.....	70
	.....	75
	.....	76
	.....	80
	.....	90

, , ,

–

–

–

DLL – Dynamic-link library

RMSE – Root Mean Square Error

MSE – Mean Square Error

PCA – principal component analysis

PE – Portable Executable

SVM – Support Vector Machine



- ,  
;  
- ,  
;  
- ;  
- ;  
:  
;  
- ,  
.  
.  
.

1

1.1

- , Data Mining,  
 1992  
 ( Data  
 Mining) - ,  
 , ,  
 . ,  
 , ,  
 , :  
 - ,  
 ;  
 - , ,  
 ;  
 - ,  
 , .  
 - ,  
 « » , ,  
 .  
 - « »  
 , ,  
 .  
 :  
 - ;

- ,  
 , .  
 ,  
 , :  
 - - ,  
 , ;  
 - - ,  
 ;  
 - - ,  
 ;  
 - -  
 .  
 , ,  
 . , , , ,  
 , ,  
 .  
 - ,  
 .

Data Mining

1.1.

[3].

1.1 –

Data Mining

/	
1	;

1.1

2	;
3	( );
4	;
5	;
6	( );
7	, .5 .4;
8	, . 1, 4 5.

,

.

1.2

.

-

,

,

,

( ) [4].

,

( ).

,

[5].

,

.

:

$m + 1$

,

,  $x_0, x_1, \dots, x_m$ ,

, 0, 1, ...,  $m$ ,

,

,

$x_0 = 1$ .

:

$$y = \left\{ \left( \sum_{i=0}^m w_i x_i \right) \right. \quad (1.1)$$

(Multilayer perceptron MLP)

( ) ,

[6, 7].

[8],

[9], [10]

[11].

[12],

[9]

[9, 5].

### 1.3

» [12]).

[13].

#### 1.3.1

(principal component analysis PCA) -

[14].

,  $x_i$  - i-

$$\Sigma = \frac{1}{m} \sum_{i=1}^m (x_i x_i^T). \tag{1.2}$$

$p$

( )

:

$$U = (u_1 | u_2 | \dots | u_p), \tag{1.3}$$

$u_1, u_2, \dots, u_p$  -  $p$

$$x^p = U^T x,$$

$p$ .

PCA

[14]

, PCA -

« »

### 1.3.2

(Independent component analysis ICA)

ICA

ICA. ICA PCA ICA [15], « ».

$$x = (x_1, \dots, x_m)^T, \quad s = (s_1, \dots, s_m)^T.$$

$$s = W x, \quad F(s_1, \dots, s_n).$$

$$x_i = (x_1, \dots, x_m)^T, \quad s_k, k = 1, \dots, n: x_i = a_{i,1}s_1 + \dots + a_{i,k}s_k + \dots + a_{i,n}s_n,$$

$a_{i,k}$ , ICA -

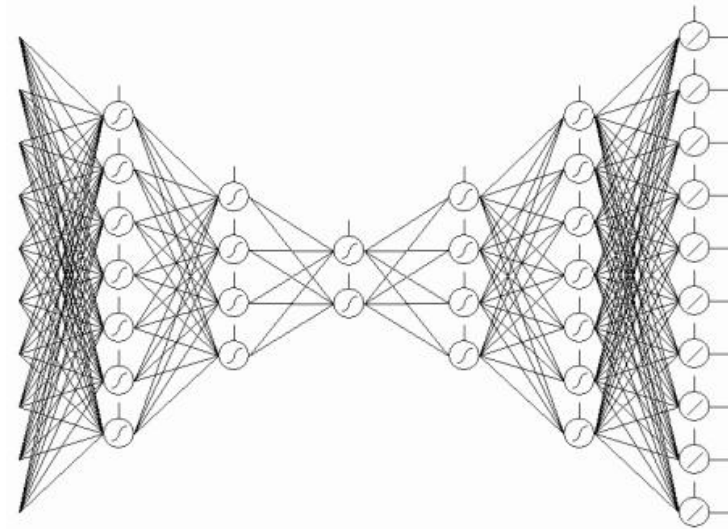
PCA.

### 1.3.3

( ) [16-17].

. 1.1. ,

« » [ ,



1.1 –

:  $g$   $f$ .

:  $h = g(x)$ ,

:  $x = g(h)$ .

,  $f$   $g$  ,

$x = f(g(x))$ ,

$$L(x, f(g(x))) \tag{1.4}$$

$$X = f(g(X)) \tag{1.5}$$

$g$   $f$

[18 - 19],

[22]

$\tilde{x}$  ,

$x$ ,

$x$ ,

( ) ,

( ).

,

,

,

.

### 1.3.4

,

,

,

-

[24, 25].

,

.

,

,

,

.

,

,

,

-

.

,

,

,

,

(

,

«

»

,

,

).  
,

,

.

(

,

)

{0,1}.

*i*

(

).  
*m* × *n*,

*W*

*W<sub>ij</sub>*

,

*h<sub>ij</sub>*

*v<sub>i</sub>*

*a<sub>i</sub>* (

)

*b<sub>j</sub>*

( ).

$(\mathbf{v}, h)$ :

$$E(\mathbf{v}, h) = -\sum_{i=1}^m a_i v_i - \sum_{j=1}^n b_j h_j - \sum_{i=1}^m \sum_{j=1}^n v_i W_{ij} h_j \tag{1.6}$$

,  
 ,  
 $(\mathbf{v}, h)$  .

$$P(\mathbf{v}, h) = \frac{1}{Z} e^{-E(\mathbf{v}, h)} \tag{1.7}$$

P - ,  $e^{-E(\mathbf{v}, h)}$   
 $h$ .  
 $h$  ,

### 1.3.5

[25].

, ,  
 ,

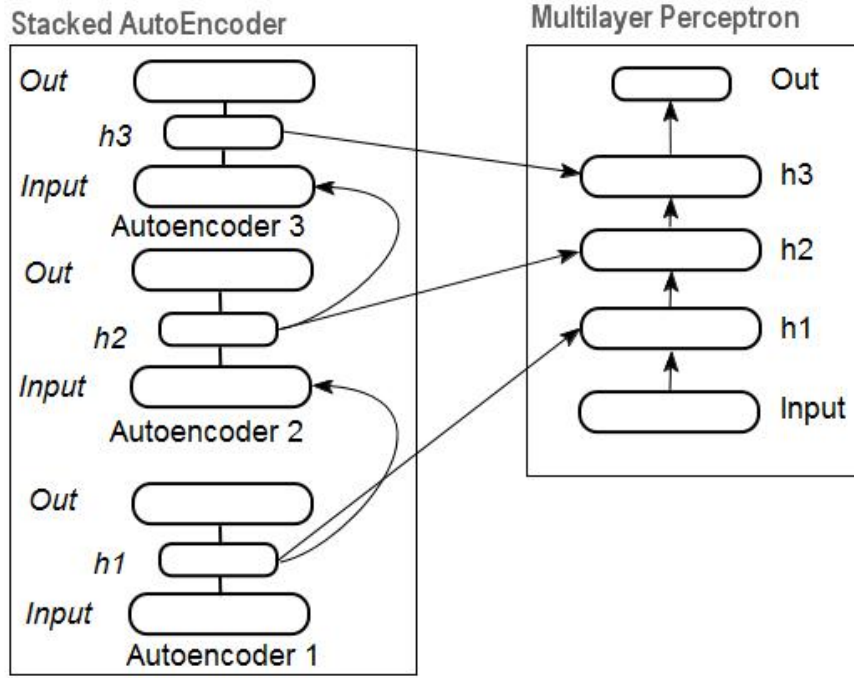
[9].

(Stacked Restricted Boltzmann Machine SRBM), (Stacked Autoencoder SAE).

Eigenfaces,

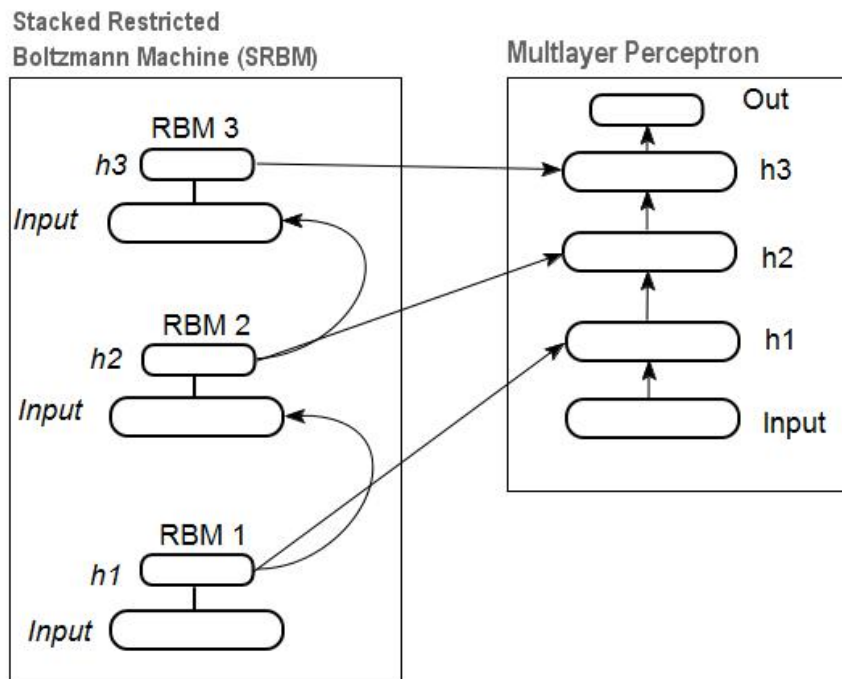
. 1.2

SAE («Deep neural network with weights initialized by Stacked AutoEncoder»).



1.2 –

SAE



1.3 –

SRBM

1.3

(SRBM)

,

SRBM («Deep neural

network with weights initialized by SRBM»).

,

,

,

,

.

2.1

, ,  
 ,  
 ( , ), , -  
 , .  
 ( ) -  
 ,  
 , ,  
 ( ) . ,  
 - ,  
 « » IoT.  
 , .  
 ,  
 [35, 36]. ,  
 1980 , « ».  
 , ,  
 , .  
 1995 , « ' » « » ,  
 .  
 . 2000



2.2

EXE -

.COM

.EXE.

Globe,

1992 .

[36].

EXE,

Windows

.COM

EXE.

(JMP)

EXE, NE, PE

ELF,

Spyware

UNIX-

Adware –

Ransomware -

## 2.3

### 2.3.1

NOP,





«IBM's T.J. Watson Research Center »

[22].

[23],

4,301

Windows,

McAfee VirusScan,

TP

FP

2.1.

2.1 –

	FP	FP	(%)
Signature + HexDump	0.34	0.00	49.32
SVN + DLLs	0.58	0.09	83.61
SVN + DLL, ,	0.71	0.08	89.36
SVN + DLL,	0.53	0.05	89.07
Naive Bayes + strings	0.97	0.04	97.11
Voting Naive Bayes + HexDump	0.98	0.06	96.88

[23],

IoT,

[24]

API

(convolutional neural network CNN)

(recurrent neural network RNN).

4,5

2

0,358%,

( )

(TP)

97%.

[25]

, RNN -  
(echo state networks ESN) API  
. (TP) 98,3% - (FP)  
- 0,1%.  
[26]

API  
(deep belief network DBN) SVM  
, ( )  
).  
98.6%.  
[27]

(SAE).  
API, Portable Executable (PE).

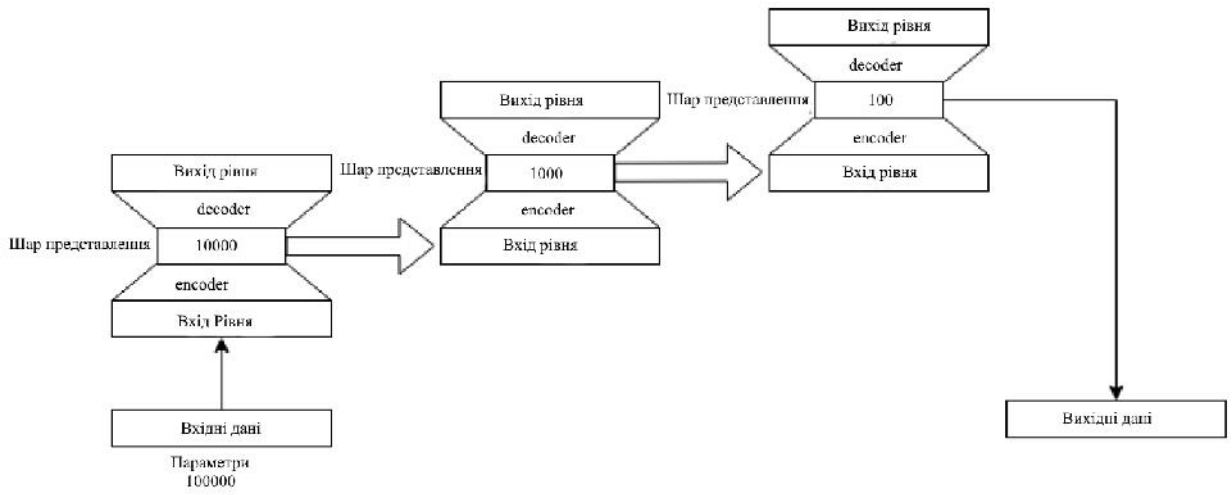
, ,  
« » , ,  
, ,  
,





3.1.

$g: X \rightarrow Y,$   
 $x_n \in X,$   
 $n,$   
 $y_m \in Y,$   
 [30].



3.1 –

SAE

$y_m\}$ .

$$D = \{\{x_0, y_0\}, \dots, \{x_m, y_m\}\},$$

$$h: X \rightarrow Y,$$

$g$

$D$ .

$h(x)$

( )

, 2

5

2

$n // 2 + 1,$   $// -$   
 $L_0, L_1, \dots, L_c;$   
 $\dim (L_c^{n // 2 + 1}),$   
 $d_i (L) - ;$

$L_i^e, L_i^r, L_i^d,$   
 $: L_i^{e_1}, \dots, L_i^{e_m}, L_i^r, L_i^{d_1}, \dots, L_i^{d_m},$   
 $Lin_{lc} = 2m + 1;$

$f (L_i),$

- a)
- b)
- c)
- d)
- e)
- f)

$f (L_i)$

$f$

$$\dim(L_0) = \text{len}(X), \tag{3.1}$$

$\text{len}(X) -$

$$\dim(L_i) = fR(L_{i-1}). \tag{3.2}$$

:

$$fR(L_i) = \frac{\dim(L_i^0)}{c}, \tag{3.3}$$

$c -$ ,  $\dim(L) -$   
 $L_i^0 -$  ;

$$\begin{cases} c * 1000, & \dim(L_i^0) > c * 1000 \\ \dim(L_i^0) - c * 100, & \dim(L_i^0) > c * 100 \\ \dim(L_i^0) - c * 10, & \dim(L_i^0) > c * 10 \end{cases} \tag{3.4}$$

$c -$ ,  $\dim(L) -$   
 $L_i^0 -$  ;

$$fR(L_i) = \dim(L_i^0) - c, \tag{3.5}$$

$c -$ ,  $\dim(L) -$   
 $L_i^0 -$  .

4.

,

(Per-Layer Reduction PLR),

$$PLR = n_{layers/2} \sqrt{\frac{\dim(L_i^0)}{fR(L_i)}} \quad (3.6)$$

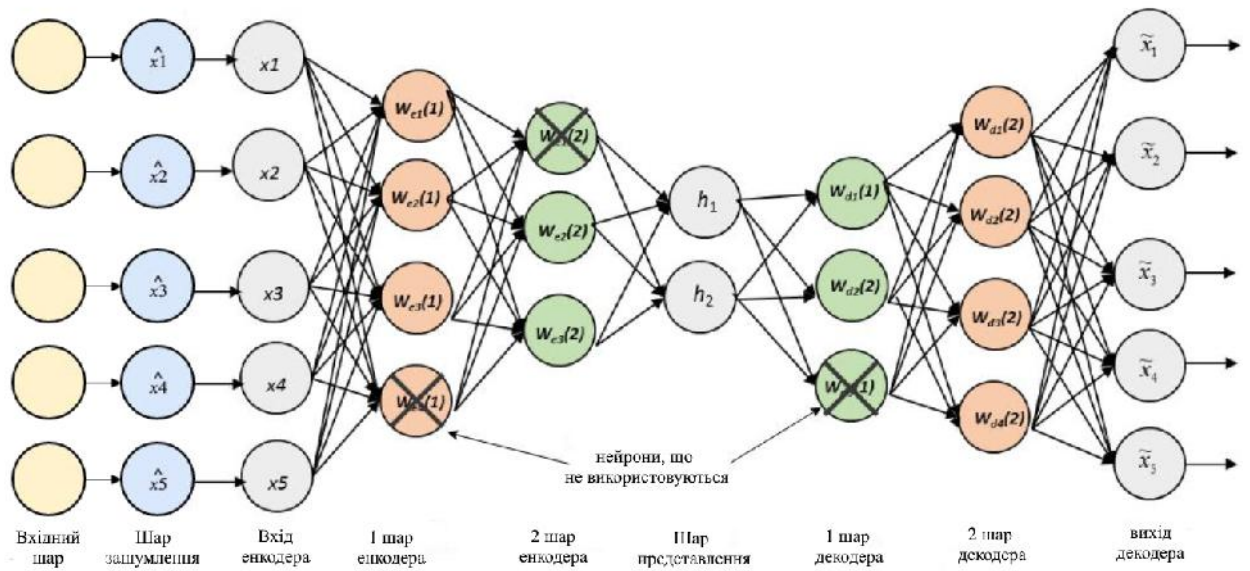
,  $di(L)$  -

,  $n_{lc}$  -

,  $L_i^0$  -

3.2

3.2.



3.2 – С

(1.5),

( )



3)

,

,

.

;

4)

,

,

.

<<

>>

.

3.3

,

.

,

,

.

,

,

.

-

(

features extraction).

,

.

,

,

:

-

;

-

;

- ( );

- N- ;

- .

-

,

.

PE-

:

,

( , ,

),

.

N- ( . Opcode - operation code,

) - ,

( ).

.

,

i .

3.4

$$P_i = \{P_i, V_i\}. \quad (3.7)$$

$$P(F) \stackrel{\text{def}}{=} (E \circ P)(F), \quad (3.8)$$

$P$  - ,

$E$  , ,

$$V = \{P_1, P_2, \dots, P_n\} \quad (3.1)$$

$$P = \{P_1, P_2, \dots, P_n\}$$

$$P = \{P_i, R_i\}$$

$$P = \{ \{P_1, V_1\}, \{P_2, V_2\}, \dots, \{P_n, V_n\} \} \quad (3.8)$$

3.1.

1.  $f_{i=1}^n t_n d$
2.  $P = P_{,1}$
3.  $V_i = P_{i,2}$
4.  $B ha = P (F)$
5.  $P = V (B ha)$
6.  $e$
7.  $S = \{ \{1, P_1\} \{2, P_2\} \dots \{n, P_{en}\} \}$
8.  $r_{Si}$

-

( ) .

,

,

,

,

.

:

*F*,*P* .

:

*S* .

,

.

4.1

```

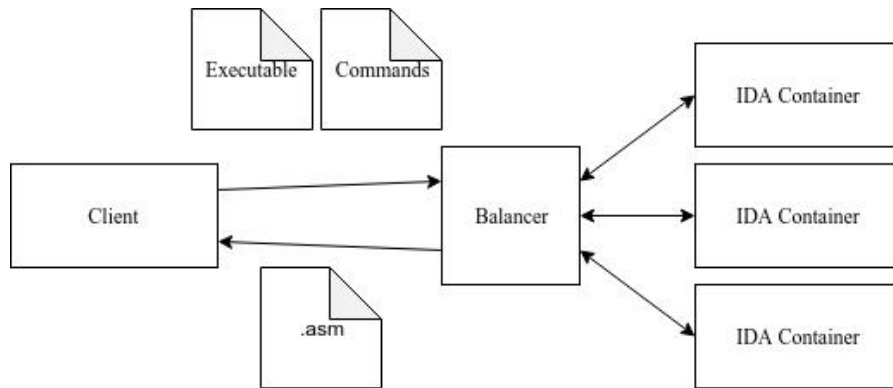
,
:
- a ;
-
, ,
), , ;
- ,
;
- ,
.
, - ,
, - .
( ),
.
, -
IDA PRO
.
IDA .asm .
, , ,

```

. IDA

IDA PRO,

. 4.1.



4.1 -

Docker -

Windows «WINE»

IDA PRO

IDA,



(control flow graph CFG) -

a.

ASM-

. 4.3

IDA PRO

API,

. 4.2,







RMSE (Root Mean

Square Error).

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - y_i^*)^2}{n}} \tag{4.3}$$

MSE

RMSE

3)

(

$$R\left(\sum_{i=1}^n w_{i,j} x_i\right), \tag{4.4}$$

R - ReLU (rectified linear unit),  $u_i - i$ ,  $x_i -$

ReLU

ReLU,

ReLU

4)

(

$$\dagger \left( \sum_{i=1}^n w_{i,j} x_i \right), \tag{4.5}$$

5) «  $\dots$  » ,

$A(y^*) - d_i - x \quad d_{y^*} -$  Dropout :

$$f(y^*) = D \odot A(y^*), \tag{4.6}$$

$D = (X_1, \dots, X_{y^*}) - d_{y^*} - X_i$  :

$$\begin{cases} p, & \text{if } k=1 \\ 1-p, & \text{if } k=0 \end{cases} \tag{4.7}$$

$k - p = P(k = 1),$

Dropout  $i-$  :

$$O_i = X_i a \left( \sum_{k=1}^{d_i} w_k x_k + b \right), \tag{4.8}$$

$q,$

$e$

$q.$

:

:

$$O_i = X_i a \left( \sum_{k=1}^{d_i} w_k x_k + b \right), \tag{4.9}$$

:

$$O_i = X_i q a \left( \sum_{k=1}^{d_i} w_k x_k + b \right), \tag{4.10}$$

$Z -$

,

$$Z_j = \sum_{i=1}^n w_{i,j} x_i, a_i = A(Z_j)$$

$A -$

$i-$

$ij:$

$$\frac{\partial RMSE}{\partial w_{i,j}} = \frac{\partial RMSE}{\partial a_j} \frac{\partial a_j}{\partial z_j} \frac{\partial z_j}{\partial w_{i,j}} \tag{4.11}$$

$i$  -  $j$   $l$ ,  
 6)  $i$  - ( )

:

$$\Delta w_{jl} = -ru_j \tag{4.12}$$

7)

)

$R$   $E$ .

:

-  $R$  =  $R$  -  $R$   $< l$ ,

-  $y^* - y < -$

1.

4.4

Python

Python

Keras. -

Python.

Deeplearning4j, TensorFlow Theano.

, : NumPy, pandas, matplotlib, SciKit Learn h5py.

NumPy

Python,

API

Pandas - Python,

API. Pandas

Python

SOLID.

,  
 . ,  
 , :  
 ;  
 - ,  
 -  
 4.2;  
 ;  
 ;  
 ;  
 ;  
 ;  
 ;

Keras,

API Keras.

API

3.4

(3.8)

$$Paterns = \left\{ \begin{array}{l} \{AsmExt, AsmRed\} \\ \{GraphExt, GraphRed\} \end{array} \right\} \tag{4.13}$$



9

, 20-

,

,

( ).

,

,

,

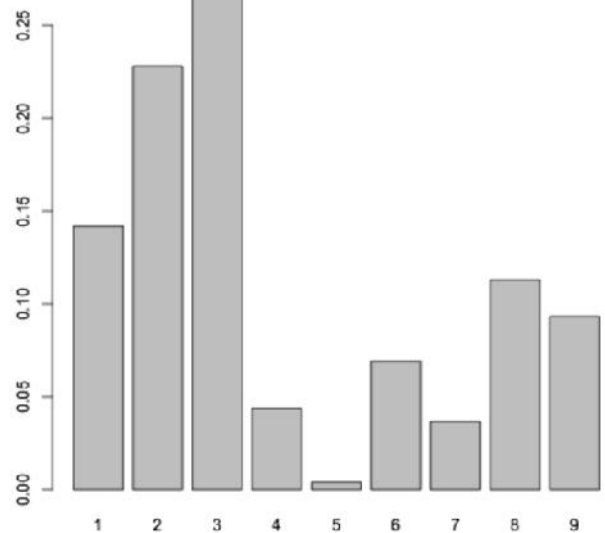
IDA.

,

,

.4.4.

	Category	Count
1	Ramnit	1541
2	Lollipop	2478
3	Kelihos_ver3	2942
4	Vundo	475
5	Simda	42
6	Tracur	751
7	Kelihos_ver1	398
8	Obfuscator.ACY	1228
9	Gatak	1013
	Total	10868



Ramnit -  
 Autorun.Inf

Lollipop - AdWare-

Kelihos -

WinPcap

45 000

4

Vundo (MS Juan)

Vundo, Virtumonde

Virtumondo,

[1].

Simda -

HTML- ' .

Tracur - ' , .

Obfuscator.ACY - «Obfuscator»

Gatak 2012

10869

10874

9

- ;
- AdWare - ;
- ;
- ;

- , .

4.6

, :

-

;

-

,

.

,

,

:

-

: 10868 , 1006

;

-

,

: 10868 ,

10854

.

,

,

,

,

.

, :

-

;

-

;

-

;

-

.

,

,

.

50.

= 4.

« », (3.3),

m = 6, 3 - , 3 - 1

m = 2;

dim (Ll ) = 50;

(3.3),

c = 4;

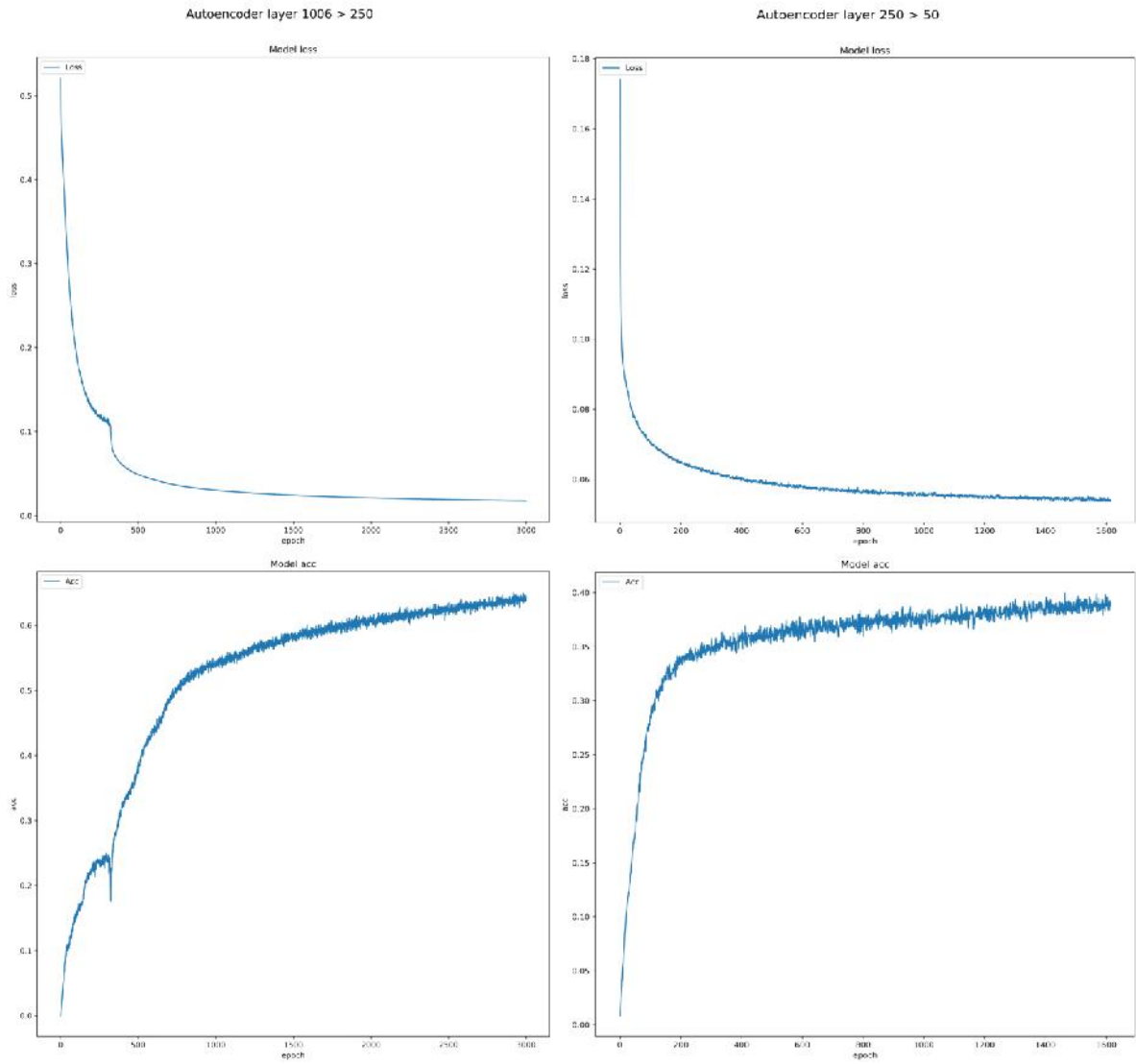
m = 6.

1006 > 250 > 50.

3000

60 %.

4.5.



4.5 –

4

10854> 2712> 6777>

168> 50.

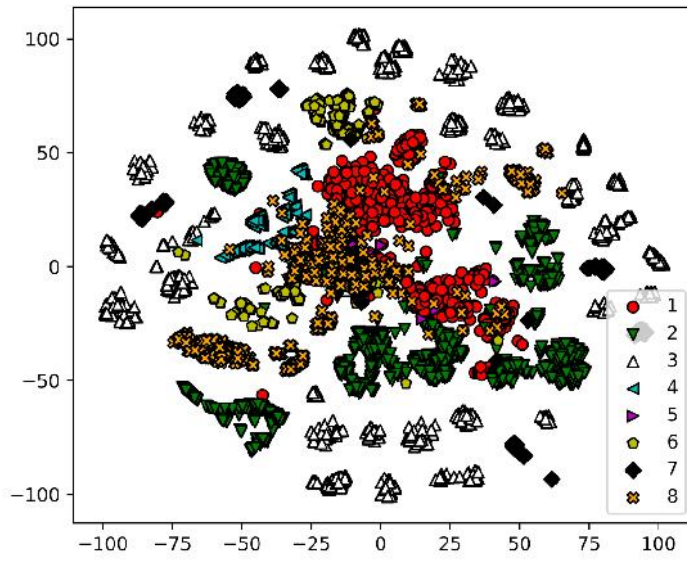
500,

) 40%.



4.2,

- :
- (PCA);
- (ICA).



4.6 –

- , :
- k (KNeighbors);
- (Decision Tree);

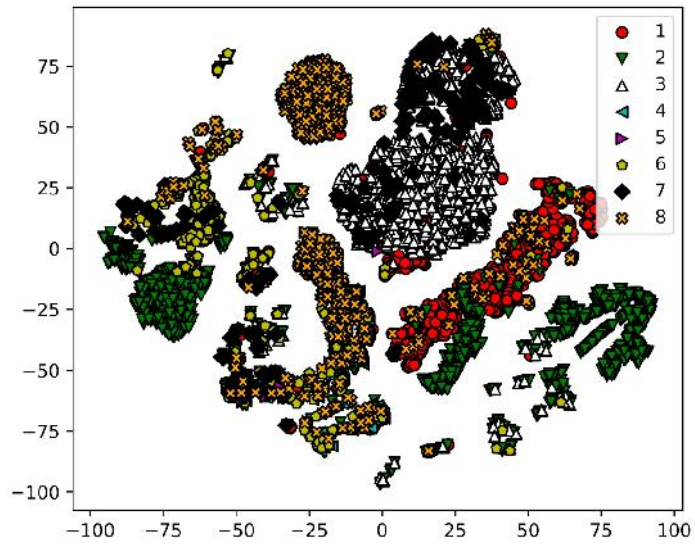
- (Random Forest);
- (MLP);
- (LDA);
- (SGD);
- (RidgeClassifierCV).

,

Scikit-Learn.

«AsmStats»

4.6 - 4.8.



4.7 –

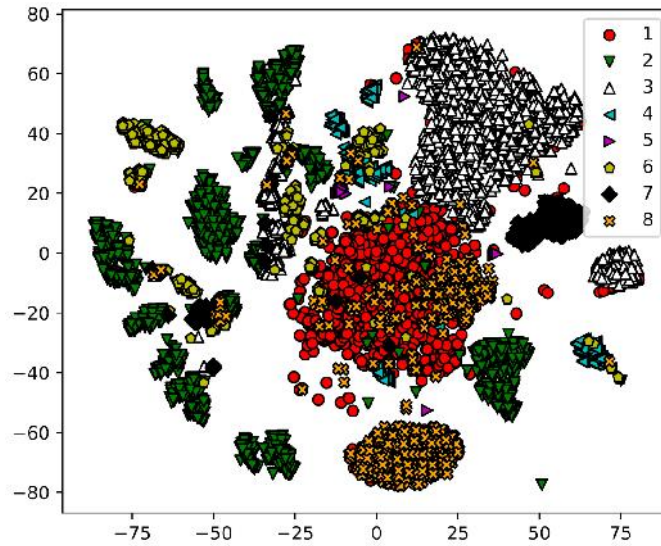
,

,

.

,

4.1.



4.6 –

4.1 –

	Transforming time,	Learning time,	Accuracy	Precision	Recall	f1-score
-	00.290277	02.343170	0.939± 0.02	0.867± 0.05	0.839± 0.04	0.844± 0.04
PCA	00.338247	02.375980	0.689± 0.28	0.667± 0.25	0.600± 0.27	0.592± 0.29
ICA	00.588323	02.236536	0.781± 0.15	0.737± 0.15	0.623± 0.23	0.641± 0.22

k ( 97%, F1  
 , 0.911),  
 ICA (

95%, F1 , 0.95).

,

,

, SGDClassifier RidgeClassifierCV,

,

.

(F1- ~ 0.3 PCA ICA ~ 0.8 ).

,

,

,

,

.

,

.

.

;

,

,

,

,

,

.

.

,

,

,

,

,

.

,

,

API

.

,

,

,

.

1. 2019 Internet Security Threat Report, ISTR Volume 24 [ ] // Symantec - 2019, - : <https://www.symantec.com/content/dam/symantec/docs/reports/istr-24-executive-summary-en.pdf> -
2. Mackinnon, Murray J., and Ned Glick. "Applications: Data Mining and Knowledge Discovery in Databases—An Overview." *Australian & New Zealand Journal of Statistics* 41.3 (1999): 255-275.
3. , . , . . . " ." 2001.
4. . , . . - , , , 3.
5. Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
6. Rui, Yong, Thomas S. Huang, and Shih-Fu Chang. "Image retrieval: Current techniques, promising directions, and open issues." *Journal of visual communication and image representation* 10.1 (1999): 39-62.
7. Romero, Diego J., Leticia Seijas, and Ana Ruedín. "Directional continuous wavelet transform applied to handwritten numerals recognition using neural networks." *Journal of Computer Science & Technology* 7 (2007).
8. LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11 (1998): 2278-2324.
9. Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems* (pp. 2172-2180).
10. Agarwal, S., & Roth, D. (2002, May). Learning a sparse representation

for object detection. In European conference on computer vision (pp. 113-127). Springer, Berlin, Heidelberg.

11. Murphy-Chutorian, E. Head pose estimation in computer vision: A survey. / E. Murphy Chutorian, M.M. Trivedi // Pattern Analysis and Machine Intelligence, IEEE Transactions on 31.4. –2009. –C. 607 – 626.

12. Kumar, Aswani Ch. "Analysis of unsupervised dimensionality reduction techniques." Computer science and information systems 6.2 (2009): 217-227.

13. Karez A. H. A Detailed Analysis of Optical Character Recognition Technology / Karez Abdulwahhab Hamad, Mehmet Kaya // International Journal of Applied Mathematics, Electronics and Computers - 01-03, 2016 – C. 244 – 249.

14. Sugiyama, Masashi. "Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis." Journal of machine learning research 8.May (2007): 1027-1061.

15. Lee, H. Efficient sparse coding algorithms / H. Lee // Advances in neural information processing systems. –2006. –C. 801 – 808.

16. Fukushima K. Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position / Kunihiro Fukushima // Biol. Cybernetics 36, 193 202 (1980) - C. 194 – 201.

17. Hinton, G.E. A practical guide to training restricted Boltzmann machines / G.E. Hinton // Momentum. –2010. –no. 9(1). – . 926.

18. Alain G. What regularized autoencoders learn from the data generating distribution. / G. Alain, Bengio // arXiv:1211.4246v5 [cs.LG] 19 Aug 2014. C. 2 – 8.

19. Skoudis E. Malware: Fighting Malicious Code. / E. Skoudis and L. Zeltser // Prentice Hall PTR 2003. – C. 208 – 250.

20. Mathur K. A Survey on Techniques in Detection and Analyzing Malware Executables / K. Mathur, S. Hiranwal // International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, Issue 4, April 2013.

21. Cock J. Computer Viruses and Malware. – Springer Science+Business

Media, LLC 2006 – C. 18 – 302.

22. Kephart J. O. Biologically inspired defenses against computer viruses. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence / J. O. Kephart, G. B. Sorkin, W. C. Arnold, D. M. Chess, G. J. Tesauro, and S. R. White //, San Francisco, CA, 1995. Morgan Kaufmann. – C. 980 – 996.

23. Su J. Lightweight classification of IoT malware based on image recognition, in Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications / J. Su, V. D. Vasconcellos, S. Prasad, S. Daniele, Y. Feng, and K. Sakurai // Conference (COMPSAC), Tokyo, Japan, July 2018. C. 664 – 669.

24. Huang W. MtNet: a multi-task neural network for dynamic malware classification, in Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, / W. Huang and J. W. Stokes // vol. 9721 of Lecture Notes in Computer Science, Springer International Publishing, Cham, Switzerland, 2016. – C. 399 – 418.

25. Thomas A. Malware classification with recurrent networks, in Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), / R. Pascanu, J. W. Stokes, H. Sanossian, M. Marinescu, and A. Thomas // IEEE, Australia, April 2014. - C. 1916 – 1920

26. David O. E. DeepSign: deep learning for automatic malware signature generation and classification, in Proceedings of the International Joint Conference on Neural Networks / O. E. David, N. S. Netanyahu, // IJCNN '15, July 2015. – C. 1 – 8.

27. William H. DL4MD: A deep learning framework for intelligent malware detection, in Proceedings of the International Conference on Data Mining (DMIN),/ H. William, C. Lingwei, H. Shifu, Y. Yanfang, and L. Xin, // The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2016. – . 61. Hamrock J Using Entropy Analysis

28. Masking wrong-successor Control Flow Errors employing data

redundancy / Javad Yousefi ; Yasser Sedaghat ; Mohammadreza Rezaee // 2015 5th International Conference on Computer and Knowledge Engineering (ICCKE). – C. 201 – 205.

29. Sereena K. Malware analysis using system call monitoring / Bai B. Mathura, K. Sereena // International Journal of Electrical, Electronics and Computer Systems (IJEECS) 2347-2820, Volume -3, Issue-11 2015. – C. 14 – 16.

30. Martovytskyi, V., Ruban, I., Bukin, I., Smyrnov, L. (2020). The model retrieves software behavior information using a hierarchical model of nested auto-associating neural networks. ’

. – C. 6– 8