Multi-Layer Model and Training Method for Information-Extreme Malware Traffic Detector

Viacheslav Moskalenko [0000-0001-6275-980], Alona Moskalenko [0000-0003-3443-3990], Artur Shaiekhov [0000-0003-3277-0264], Mykola Zaretskyi [0000-0001-9117-5604]

Sumy State University, Rimsky-Korsakov st., 2, Sumy, 40007, Ukraine
v.moskalenko@cs.sumdu.edu.ua, a.moskalenko@cs.sumdu.edu.ua

Abstract. Model-based on multilayer convolutional sparse coding feature extractor and information-extreme decision rules for malware traffic detection is presented in the paper. Growing sparse coding neural gas algorithms for unsupervised pre-training of the feature extractor are used. Random forest regression model as a student in knowledge distillation from sparse coding layers is proposed for speed up inference mode. Information-extreme learning method based on binary encoding with tree ensembles and class separation with radial basis function in binary Hamming space are proposed. Information-extreme classifier is characterized by low computational complexity and high generalization ability for small labeled training sets. Simulation results with an optimized model on test open datasets confirm the suitability of proposed algorithms for practical application.

Keywords: malware detection system, convolutional sparse coding network, growing neural gas, tree ensembles, random forest regression, information criterion, information-extreme machine learning.

1 Introduction

Existing malware traffic detection systems still do not provide high-reliability solutions, as there are a constant increase the number and variety of new sources of malware traffic and a small number of relevant labeled data [1, 2]. Thus, the use of handcrafted features for the description of observations leads to a decline the informativeness of the features description and the effectiveness of learning of the decision rules of the malware traffic detection system [2, 3]. Therefore, the most promising approach to the synthesis of a features extractor is the use of ideas and methods of machine learning for the hierarchical (deep) representation of observations for unlabeled data [4, 5].

Conventional approaches to deep supervised machine learning require a significant amount of labeled training examples and computational resources [6, 7]. In addition, models trained with a supervisor based on gradient descent and its modifications are vulnerable to adversarial attacks, noise and data novelty. To increase the informativeness of the feature representation of observations, it is promising to use ideas and methods of sparse coding and unsupervised competitive learning [8, 9]. This allows to use the large volume of unlabeled data with maximum efficiency. Among the ways to

increase the generalization ability of the decision rules are considered ensemble algorithms, error correction codes and methods of class separation within the geometric approach. Also high speed of packet flow in modern networks require high productivity of traffic analysis algorithms. To reduce computational complexity of data analysis models, different methods of model pruning and knowledge distillation are used. However, models hybridization and integrated use of different methods bring some uncertainties to the final result, so the solution in this approach requires research and verification. In this case, information criteria are considered the best metrics for validation and verification of the result, because they directly characterize the reduction of uncertainty in decision-making and are less sensitive to outlier and imbalances in the data.

2 Formal Problem Statement

Let the CTU-Mixed and CTU-13 datasets are given data collections from the real network environment by CTU researchers from 2011 to 2015, which are formed as pcap-files [4, 5]. The first CTU-Mixed dataset can be used for training a feature extractor. The second CTU-13 dataset contains labeled flows and it could be used to train the decision rules for detecting malware network traffic.

It is necessary to build an informative feature extractor and reliable decision rules using labeled and unlabeled datasets through optimization of model parameters. In the process of training, it is necessary to maximize the information efficiency criterion of the malware traffic detector

$$\overline{E}^{*} = \frac{1}{M} \sum_{m=1}^{M} \max_{\{k\}} E_m^{(k)},$$

where $E_m^{(k)}$ is information efficiency criterion of recognition the class X_m^o on k-th step of training; $\{k\}$ – ordered set of training steps.

When the malware traffic detector functions in its inference mode, it is necessary to provide computational efficiency for high speed traffic.

3 Literature Review

Convolutional multi-layer neural networks allow forming an informative hierarchical features representation of input observations [6]. In addition, they have already shown high efficiency in solving problems of machine vision and analysis of time series [6], [7]. Meanwhile, supervised training requires a large amount of labeled data, the labeling of which may be expensive or inaccessible in a reasonable amount of time. The unsupervised training of convolutional networks is aimed at efficient use of unlabeled examples, which are usually available quite a lot. It is carried out based on an autoencoder or Restricted Boltzmann machine, which requires a large amount of training

data and long learning time to obtain an acceptable result [8]. In work [9] it is proposed to use alternative approach based on k-means cluster-analysis algorithm to speed up feature set training. However, k-means is characterized by slow convergence and sub-optimality of the results due to the hard-competitive nature of its learning scheme and the sensitivity to initial cluster initialization.

In work [10] is proposed a combination of the principles of neural gas and sparse coding for the feature set training on unlabeled data. Given approach is characterized by soft-competitive learning scheme that facilitates robust convergence to close to optimal features distributions over the training sample. At the same time, embedding of sparse coding methods can increase the immunity against interference and generalization ability of features representation. Also, it is a well-known fact that sparse representations of the input data are a crucial tool for combating adversarial attacks and the production of de-correlated features as a result of the explaining-away effect. However, the size of feature set is unknown beforehand and it is selected by the developer, which leads to increase the optimization time.

The required size of feature set in each layer of hierarchical representation is difficult to predict in advance, so the promising approach to feature set learning is to use the principles of growing neural gas, which automatically determines the required number of neurons (features) [11]. The presence of a mechanism for the adding of new neurons, as well as the removal of excessive old ones, makes the algorithm more flexible compared to the classical neuron gas, but it also has serious disadvantages. The small values of the period between the iterations of the generation of new neurons λ lead to the instability of the learning process and the distortion of the formed structures, as here observed the excessively frequent adding of new neurons. The high value of the period λ provides the expected effect, but at the same time it leads to a significant slowdown in the algorithm. However, in the works [11, 12] it was shown that achieving stability of learning could be done by setting the "radius of reach" of the neurons, which involves the replacement of the parameter λ on the threshold of maximum distance of the neuron from each points of the training set attributed to it. However, the mechanisms for updating neurons and assessing the remoteness of the points of the input space to the neurons have not yet been reviewed in order to adapt the learning process to the sparse coding of observations.

The main disadvantage of sparse coding in representation learning is the use of an iterative procedure during the inference which slows down the recognition process. One of the popular ways to accelerate models is to use the principles of knowledge distillation, where the redundant model acting as a teacher can be replaced by a lightweight model acting as a student [13]. The ensemble of decision trees is a flexible and computationally efficient model, which can potentially be used as a student model to approximate the sparse coder [14]. However, no such research had been conducted and the effectiveness of such an approach is unknown, which underscores the relevance of this issue.

In addition the decision rules are important components in the malware detection systems. As a rule, it represents a trainable classifier. At the same time, the effectiveness of training a classifier is often considered as a measure of the effectiveness of the feature extractor [5]. The most popular algorithm for classification analysis is the

method of support vector machine, where the training of decision rules takes place within the framework of a geometric approach by constructing linear separable hyper-surface in the secondary features space [15]. However, this algorithm requires a lot of hyper-parameters adjustments and its performance depends on the complexity of the kernel functions. In work [16], were proposed the construction of decision rules by adaptive binary encoding of the input features and the optimizing in information sense the radial-basis based separable hyper-surface in the Hamming binary space. Such a classifier has high operational efficiency, since it uses low computing complexity operations as comparison and logical XOR.

4 Model and Training Method for Malware Traffic Detector

The internal characteristics of the unit of traffic (packet stream or session) are best displayed in the front part of its bytes, which contains connection data and some content data. The process of converting a pcap-file into a training data set involves three main steps: the separation of traffic into discrete units, taking into account some granularity, clearing traffic by removing empty and duplicate units, forming training images. When dividing traffic into discrete units, one can consider the following granularities: TCP connection, flows, session, service, and host. In this paper, it is proposed to divide the incoming traffic into flows, where a number of packets have the same tuple of five elements: the source and destination IP address, the source and destination ports, the protocol number. In this case, the length of the stream is limited to 784 bytes, so longer streams are cropped, and shorter ones are supplemented by zero bytes. As a result, we have an image of 28x28 pixels, which will be delivered to the input of the feature extractor. The brightness of each pixel is normalized to the range [0, 1].

As a basis for building architecture of features extractor was used a convolutional network is known as LeNet-5 [5], the main modification of which relates to use the unfixed number of convolutional filters, the amount of which is determined during the layer-wise training. The pixel activation of each channel of features map is offered to calculate based on greedy-L0 Orthogonal Matching Pursuit algorithm (OMP) or L1-regularized least angle regression algorithm (LARS) with the function of ReLU activation [17]. In order to accelerate the model in the inference mode, it is possible to replace the computationally intensive search for sparse coefficients with a non-iterative approximating encoder (Figure 1). According to distillation knowledge principle, the training set for approximation encoder is formed from input of the layer and pseudo-labels from output of the layer. In this case, pseudo-labels are obtained by OMP or LARS algorithms.

It is proposed to implement sparse coding with OMP and LARS algorithms where stop criterion based on achievement of 30% non-zero entries in sparse code. A Local Contrast Normalization layer, placed after the sub-sampling layer, before the next layer, amplifies the informative features and weakens the rest of the pixels of the feature map.

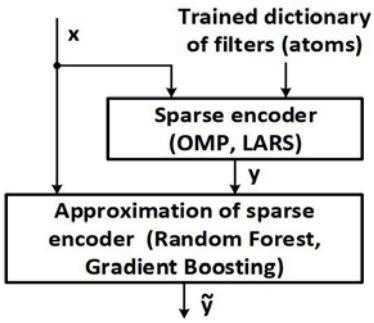


Fig. 1. Knowledge distillation diagram for each layer of feature extractor

The dataset for training of feature extraction layer is formed by decomposition of images or activation maps to patches. These patches are reshaped to 1D vectors, which put on the input of growing sparse coding neural gas algorithm, main steps of which are given below [16].

1. Initialization of the counter of training vectors $t := 0$.
2. Two initial nodes (neurons) w_a and w_b are assigned by random selection from the training set. Nodes w_a and w_b are connected by an edge whose age is zero. These nodes are considered non-fixed.
3. Selected from the dataset the following vector x, which is normalized to a unit length (L2-normalization).
4. Normalizing each base vector $w_k, k = \overline{1, M}$ to a unit length (L2-normalization).
5. Calculation of the similarity of the input vector x to the base vectors $w_{s_k} \in W$ for their sorting

$$-(w_{s_0}^T x)^2 \le \ldots \le -(w_{s_k}^T x)^2 \le \ldots \le -(w_{s_{M-1}}^T x)^2 .$$

6. The closest node is selected w_{s_0} and the second closest to the node w_{s_1}.
7. Increase the age of all incident edges w_{s_0} by one.
8. If w_{s_0} is fixed, then should move to step 9, otherwise, to step 10.
9. If $(w_{s_0}^T x)^2 \ge v$, then proceed to step 12. Otherwise, should be added a new non-fixed neuron w_r to a point that coincides with the input vector $w_r = x$, also is adding a new edge, that connects w_r and w_{s_0}, then proceed to step 13.
10. The node w_{s_0} and its topological neighbors (the nodes connected to it by the edge) are displaced in the direction to the input vector x by the next formulas [10]

$$\Delta w_{s_0} = \varepsilon_b \eta_t y_0 (x - y_0 w_{s_0}), \; y_0 := w_{s_0}^T x,.$$

$$\Delta w_{s_n} = \varepsilon_n \eta_t y_n (x - y_n w_{s_n}), \; y_n := w_{s_n}^T x,.$$

$$0 < \varepsilon_b << 1, \; 0 < \varepsilon_n << \varepsilon_b,$$

$$\eta_t := \eta_0 (\eta_{final} / \eta_0)^{t/t_{\max}},$$

where Δw_{s_0}, Δw_{s_n} – vectors of correction of weight of the neuron-winner and its topological neighbors, respectively; ε_b, ε_n – the constants of updated forces of weighting coefficients of the neuron-winner and its topological neighbors respectively; $\eta_0, \eta_t, \eta_{final}$ – initial, current and final learning rate respectively.

11. If $(w_{s_0}^T x)^2 \geq v$, note the neuron w_{s_0} as fixed.
12. If w_{s_0} and w_{s_1} are connected by edge, then its age is reset, otherwise a new edge with a zero age is formed between w_{s_0} and w_{s_1}.
13. All edges in the graph with the age more than $a_{\max}$ are removed. In the case that some nodes do not have incident edges (become isolated), they are also removed.
14. If $t < t_{\max}$ then proceed to step 15, otherwise – increment of the counter of steps is $t := t + 1$ and then proceed to step 3.
15. If all neurons are fixed, the execution of the algorithm stops, otherwise proceed to step 3 and a new epoch of learning begins (repetition of the training set).

Features extractor can be fine-tuned based on the backpropagation algorithm with a temporary or permanent neural classifier at the model output [17]. Since in the conditions of nonstationarity the informativeness of features in advance cannot be known, the fine tuning is not provided in our algorithm. The purpose of the feature extractor is to disentangle explanatory factors.

The information-extreme classifier requires binary representation of the input signal to build error-correction decision rules. The ensemble of decision trees is a computationally effective method for inducing informative binary features of observations (Figure 2). Nodes of decision trees are numbered. Numbers of nonzero bits of resulting binary code correspond to the numbers of nodes through which the decision path lies [16].

Information-extreme classifier under inference mode make decision on belonging of input datapoint x with appropriate binary representation b to one class from set $\{X_z^o \mid z = \overline{1, Z}\}$ according to maximum value of membership function $\mu_z(b)$ through the expression $\arg\max_z \{\mu_z(b)\}$.

In this case membership function $\mu_z(b)$, the optimal container of which has support vector b_z^* with dimension N_2 and radius d_z^*, is derived from formula

$$\mu_z(b) = \exp\left(-\sum_{i=1}^{N_2} b_i \oplus b^*_{z,i} / d^*_z \right). \quad (1)$$

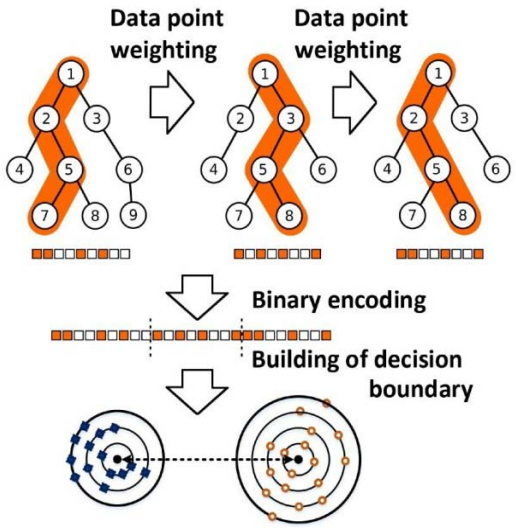


Fig. 2. Classifier Architecture

Let $D = \{x_j, y_j \mid j = \overline{1,n}\}$ be a training set, where n is size of dataset and y_j is label of *j*-th datapoint, which correspond to one class from set of classes $\{X^o_z \mid z = \overline{1,Z}\}$. In this case, classifier evaluates belonging of *j*-th datapoint x_j with N_1 features to one of the Z classes performs feature encoding using decision trees and decision rules constructed in radial basis of binary Hamming space. The training of information-extreme classifier is performed according to the following steps.

1. Initialize weight $w_j = 1/n$.
2. For $k = 1,\ldots,K$ do
3. Bootstrap D_k from D using probability distribution $P(X = x_j) = w_j$.
4. Train decision tree T_k on D_k using entropy criterion to measure the quality of split.
5. Binary encoding of x_j datapoint from D using concatenation of results from $T_1,\ldots,T_k$ trees. The output of this step is a binary matrix $\{b_{z,s,i} \mid i = \overline{1,N_2}; s = \overline{1,n_z}; z = \overline{1,Z}\}$, where N_2 is a number of induced binary features and n_z is a number of samples corresponded to class X^o_z. Hence the equality $n = \sum_z n_z$ condition is met.

6. Build information-extreme decision rules in radial basis of binary Hamming space and compute optimal information criterion:

$$E_z^* = \max_{\{d\}} E_z(d), \tag{2}$$

where $\{d\} = \{0,1,...,\left(\sum_i b_{z,i} \oplus b_{c,i} - 1\right)\}$ is a set of concentric radiuses with center b_z (support vector) of data distribution in class X_z^o, which computed using rule

$$b_{z,i} = \begin{cases} 1, & \text{if } \frac{1}{n_z}\sum_{s=1}^{n_z} b_{z,s,i} > \frac{1}{Z}\sum_{c=1}^{Z}\frac{1}{n_c}\sum_{s=1}^{n_c} b_{c,s,i}; \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

where E_z – training efficiency criterion of decision rule for X_z^o class, which is computed as the normalized modification of the S. Kullback's information measure [16]:

$$E_z = \frac{1-(\alpha_z+\beta_z)}{\log_2(2+\varsigma)-\log_2\varsigma}\cdot\log_2\left[\frac{2-(\alpha_z+\beta_z)+\varsigma}{(\alpha_z+\beta_z)+\varsigma}\right], \tag{4}$$

where α_z, β_z are the false-positive and false-negative rates of classification of input vectors as belonging to the X_z^o class; ς is any small non-negative number, introduced to avoid uncertainty when dividing by zero.

7. Test obtained information-extreme rules on dataset D and compute error rate for each sample from D. Under the inference mode, decision on belonging of datapoint b to one class from set $\{X_z^o \mid z = \overline{1,Z}\}$ is made according to maximum value of membership function $\mu_z(b)$ through the expression $\arg\max_z\{\mu_z(b)\}$. In this case membership function $\mu_z(b)$ of binary representation *b* of input datapoint *x* to X_z^o class, the optimal container of which has support vector b_z^* and radius d_z^*, is derived from formula (2).
8. Update $\{w_j\}$ proportional to errors on datapoint x_j :

$$w_j = 1 - \mu_{m'}(x_j),\ m' = y_j;$$
$$w_j = \frac{w_j}{\sum_j w_j}.$$

9. If $|E_k^* - E_{k-1}^*| < \varepsilon$ and k < K/2 abort loop, where ε = 0.001.

Thus, the resulting model consists of several layers of tree ensemble with optimal in informational sense decision rules at the output.

5 Result and discussion

The training sample formed with CTU-Mixed for the training of the feature extractor contains 10,000 instances. To train the information-extreme classifier are formed by 1000 instances per class in the training and test datasets. In growing sparse coding neural gas algorithm were chosen the following parameters $\varepsilon_b = 0.5$, $\varepsilon_b = 0.05$, $a_{\max} = 100$, $\eta_0 = 1$ та $\eta_{final} = 0.01$. The parameter of the threshold of neuron fixation v and the parameter of the maximum number of trees K of the classifier are adjusted by scrolling through the values. Table 1 shows the dependence of the number of neurons in the first M_1 and second M_2 layers of feature extractor, the criterion of the effectiveness of training averaged over the classes $\bar{E}$ and accuracy by the validation sampling of the parameter v. In the tree ensembles, max depth is set to 5 and max features is set to $\sqrt{N_1}$.

Table 1. Dependence of information criteria and number of neurons from model parameters

v	M_1	M_2	$\bar{E}$	*Validation accuracy*
0.10	15	11	0.106	74
0.15	17	13	0.138	77
0.20	23	13	0.138	77
0.25	25	13	0.138	77
0.30	27	15	0.149	78
0.35	27	15	0.220	83
0.40	33	17	0.255	85
0.45	34	22	0.255	85
0.50	40	25	0.366	90
0.55	49	31	0.459	93.0
0.60	66	43	0.466	93.2
0.65	70	45	0.501	94.1
0.70	99	45	0.550	95.2
0.75	145	57	0.554	95.3
0.80	161	120	0.591	96.1
0.85	220	147	0.603	95.4
0.90	322	238	0.611	95.0

The analysis of Table 1 shows that increasing the threshold ν leads to an increase in the number of neurons in the process of unsupervised training the features extractor. At the same time, increasing the threshold from 0.8 to 0.9 practically does not affect the accuracy of the decision rules. It means, that the value $\nu^* = 0.8$ is optimal and allows to form a more compact features representation (compression), meanwhile $\nu = 0.9$ allows to form a sparse representation based on overcomplete basis.

Knowledge distillation is implemented with Random Forest regression as student model, where the number of decision trees is limited to 150. The obtained model has equivalent accuracy. In this case, the inference time is reduced by 65 times.

Figure 3 shows a graph of maxima's changes of the information criterion (4) averaged in the set of classes in dependence on the number of decision trees in informa-

tion-extreme classifier with $v^* = 0.8$. In this case, the maximum number of trees is limited, $K = 100$.

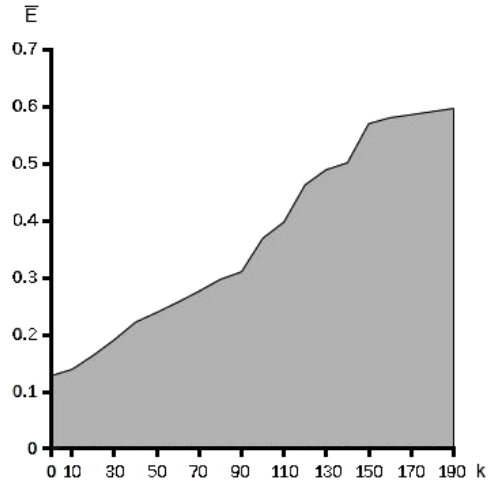


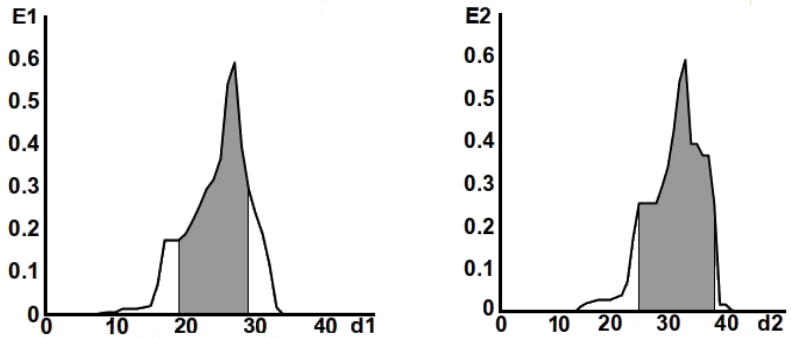
Fig. 3. A graph of the change of the average information criterion (4) in dependence from the number of decision trees in information-extreme classifier

The analysis of Figure 3 shows that the optimal value of the hyper parameter K^* is equal to 185. Further increase of the parameter K does not lead to an increase in the accuracy of the decision rules. At optimal parameters of the extractor and the classifier, the accuracy of detection of malware traffic is 96.1%. It indicates on the informative nature of the features descriptive of observation. Figure 4 shows the dependence of the information criterion (4) on the code radius of the container of each class.

The analysis of Figure 4 shows that the maximum values of information criterion of learning for the first and second classes are equal to $E_1^* = 0.590$ and $E_2^* = 0.597$, respectively, and the optimal values of radii of the corresponding containers of the classes of recognition $d_1^* = 26$, $d_2^* = 32$ (in code units). In this case, the inter-center Hamming distance is 65 indicating compactness of the feature vector distributions and the clarity of partition in the binary Hamming space.

Thus, the proposed training algorithm allows determining automatically the optimal number of neurons at each layer. At the same time, approximation of the sparse encoder by the non-iterative model, Random Forest regression, allowed accelerating the inference mode.

The results of simulation on data from CTU-Mixed and CTU-13 datasets show that obtained result is superior to result from [4] and [5] and it is acceptable for practical applications.



a b

Fig. 4. Charts of dependency of the information criterion (4) on the radii of containers of classes: a – class of normal traffic; b – class of malware traffic

6 Conclusions

10. The scientific novelty of the obtained results is as follows:

- the algorithm of growing sparse coding neural gas is proposed for the first time, which allows unsupervised learning the optimal set of neurons for each layer of the convolution sparse coding model of feature extractor model;
- for the first time it was proposed to apply the principle of knowledge distillation to reduce computational costs in the algorithms of sparse coding through the application of approximation by the random forest model, which in the inference mode is non-iterative and computationally efficient;
- for the first time, an information-extreme algorithm of supervised learning is proposed for constructing the decision rules of the detector of malware network traffic.

11. The practical value of obtained results obtained for malware traffic detection systems is a developing a new learning method that effectively uses both labeled and unlabeled training sets. The results of simulation with using the CTU-Mixed and CTU-13 datasets confirm the effectiveness of the obtained decision rules in identifying the malware in test samples of traffic. In this case, the accuracy of the decision rules of the malware traffic detector is 96.1%.

7 Acknowledgment

The work was performed in the laboratory of intellectual systems of the computer science department at Sumy State University with the financial support of the Ministry of Education and Science of Ukraine in the framework of state budget scientific and research work of DR No. 0117U003934.

References

1. Skrzewski, M.: Flow Based Algorithm for Malware Traffic Detection. Computer Networks. 271-280 (2011).
2. Berkay Celik, Z., Walls, R., McDaniel, P., Swami, A.: Malware traffic detection using tamper resistant features. MILCOM 2015 – 2015 IEEE Military Communications Conference. (2015).
3. Iglesias, F., Zseby, T.: Analysis of network traffic features for anomaly detection. Machine Learning. 101, 59-84 (2014).
4. Yousefi-Azar, M., Varadharajan, V., Hamey, L., Tupakula, U.: Autoencoder-based feature learning for cyber security applications. 2017 International Joint Conference on Neural Networks (IJCNN). (2017).
5. Wei Wang, Ming Zhu, Xuewen Zeng, Xiaozhou Ye, Yiqiang Sheng: Malware traffic classification using convolutional neural network for representation learning. 2017 International Conference on Information Networking (ICOIN). (2017).
6. Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015).
7. Zhao, B., Lu, H., Chen, S., Liu, J., Wu, D.: Convolutional neural networks for time series classification. Journal of Systems Engineering and Electronics. 28, 162-169 (2017).
8. Qiying Feng, Chen, C., Long Chen: Compressed auto-encoder building block for deep learning network. 2016 3rd International Conference on Informative and Cybernetics for Computational Social Systems (ICCSS). (2016).
9. Tang, J., Wang, D., Zhang, Z., He, L., Xin, J., Xu, Y.: Weed identification based on K-means feature learning combined with convolutional neural network. Computers and Electronics in Agriculture. 135, 63-70 (2017).
10. Labusch, K., Barth, E., Martinetz, T.: Sparse Coding Neural Gas: Learning of overcomplete data representations. Neurocomputing. 72, 1547-1555 (2009).
11. Mrazova, I., Kukacka, M.: Image Classification with Growing Neural Networks. International Journal of Computer Theory and Engineering. 422-427 (2013).
12. Palomo, E., Lopez-Rubio, E.: The Growing Hierarchical Neural Gas Self-Organizing Neural Network. IEEE Transactions on Neural Networks and Learning Systems. 1-10 (2016).
13. Li, H., Lin, S., Chen, C., Chiang, C.: Layer-Level Knowledge Distillation for Deep Neural Network Learning. Applied Sciences. 9, 1966 (2019).
14. Zhou, Y., Zhou, Z., Hooker, G.: Approximation Trees: Statistical Stability in Model Distillation, https://arxiv.org/abs/1808.07573. (2018).
15. Kim, S., Yu, Z., Kil, R., Lee, M.: Deep learning of support vector machines with class probability output networks. Neural Networks. 64, 19-28 (2015).
16. Moskalenko, V., Moskalenko, A., Korobov, A., Semashko, V.: The Model and Training Algorithm of Compact Drone Autonomous Visual Navigation System. Data. 4, 4 (2018).
17. Y. Gwon, M. Cha, H. T. Kung: Deep Sparse-coded Network (DSN). 2016 International Conference on Pattern Recognition. (2016).