Results in Control and Optimization 14 (2024) 100382

Contents lists available at ScienceDirect

Results in Control and Optimization

journal homepage: www.elsevier.com/locate/rico

Radon transform based malware classification in cyber-physical system using deep learning

Rasim Alguliyev, Ramiz Aliguliyev, Lyudmila Sukhostat *

Institute of Information Technology, 9A, B. Vahabzade Street, Baku, AZ1141, Azerbaijan

ARTICLE INFO

Keywords:
Malware classification
Cyber-physical system
Radon transform
Transfer learning
Deep neural network

ABSTRACT

The development of cyber-physical systems entails the growth and diversity of malware, which increases the scale of cybersecurity threats. Attackers use malicious software to compromise various components of cyber-physical systems. Existing technologies make it possible to reduce the risk of malware infection using vulnerability and intrusion scanners, network analyzers, and other tools. However, there is no perfect protection against the increasingly sophisticated types of malware. The goal of this research is to solve this problem by combining different visual representations of malware and detection models based on transfer learning. This method considers two pre-trained deep neural network models (AlexNet and MobileNet) that are capable of differentiating various malware families using grayscale images. Radon transform is applied to the resulting grayscale malware images to improve the classification accuracy of the new malware binaries. The proposed model is evaluated using three datasets (Microsoft Malware Classification, IoT_Malware and MalNet-Image datasets). The results show the superiority of the proposed model based on transfer learning over other methods in terms of the efficiency of classifying malware families aimed at infecting cyber-physical systems.

Introduction

In recent years, malicious software for cyber-physical systems (CPSs) has evolved [1–4]. Increased exposure to malware is one of the highest impacts faced by CPSs in connection with increased digitalization, such as cloud technology, e-commerce, and convergence of information technology (IT) and operational technology (OT) [5,6].

CPSs consist of a large number of connected devices (e.g., sensors, smart meters, etc.) that are targeted by many malware families, such as Tsunami, Bashlite, and Mirai [7]. They take advantage of weak authentication, outdated firmware, and scanners designed to find open ports and compromise system devices [4].

Malware has much in common with each other, namely similar source code [1,2]. They are constantly evolving with support for DDoS attacks to more critically infect system devices [3].

The weakest link in the CPS security chain is the human factor. Cybercriminals use this factor to gain unauthorized access, steal personal data, and infect systems with malware [8].

Targeted malware as a type of cyber weapon has the following features [9]:

- Exploitation of vulnerabilities, including zero-day attacks;

* Corresponding author.
E-mail address: lsuhostat@hotmail.com (L. Sukhostat).

https://doi.org/10.1016/j.rico.2024.100382
Received 2 May 2023; Received in revised form 20 January 2024; Accepted 22 January 2024
Available online 4 February 2024

- Disguise and self-destruction;
- Wide functionality in terms of solving target tasks;
- Infrastructure support, updating, and management;
- High quality of the code;

etc.

Among the various variants of malicious programs, one can identify crypto-miners, viruses, ransomware, worms, spyware, etc. Their main goal is illegally collecting information, blocking services, and cyber espionage.

Thus, based on the analysis of the CPS malware characteristics, we can conclude that there are static and dynamic characteristics [10,11]. They can be used as features to detect malicious code, such as function call graphs, strings, grayscale images, etc.

Malware detection depends entirely on the cyberattack vectors that can be observed on the basis of malware sample analysis. New approaches treat the executable file as a sequence of assembly language instructions (Asm file) or a sequence of bytes [12]. The malware byte file is the hexadecimal representation of the portable executable (PE) malicious file [13,14]. Features extracted from such files do not always provide helpful information for the classifier [12]. However, this is easy to see when visually analyzing the

Table 1
Overview of state-of-the-art malware classification methods.

References	Proposed approach	Main contribution	Limitations	Dataset	Methods
Nguyen et al. (2023) [28]	Generative adversarial networks for multiclass malware classification	• Comparison of different techniques for generating images from malware samples • GAN-generated "deep fake" malware images evaluation	Do not do well with malware that is more general or obfuscated	MalExe (20 families)	GAN, SVM, kNN, multilayer perceptron (MLP), RF, Restricted Boltzmann Machines, XGBoost, ResNet152
Bhodia et al. (2019) [29]	Transfer Learning for Image-Based Malware Classification	• The models are able to generalize the data • Outperformed kNN in simulated zero-day	The k-NN learning technique outperformed the model in some cases.	Malimg (25 families), Malicia (54 families)	ResNet34, ResNet50, ResNet101, ResNext50, kNN
Prajapati and Stamp (2021) [30]	An Empirical Analysis of Image-Based Learning Techniques	• ResNet152 and VGG-19 showed the best performance • Improvement in the classification accuracy of Obfuscator families	Opcode-based results performed relatively poorly	Malicia dataset (20 families) + (17 families) [24]	MLP, CNN, LSTM, RNN, GRU, ResNet152, VGG-19
Yajamanam et al. (2018) [32]	Malware score based on gist descriptors	• Analysis of this gist-based scoring technique robustness when applied to obfuscated malware • Much more efficient than relying on gist descriptors.	Limited experiments	Malimg (25 families), Malicia (8 families)	SVM, kNN, Inceptionv3
Tekerek and Yapici (2022) [26]	Malware classification and augmentation model based on CNN	A new CycleGAN-based data augmentation method was developed	Worse performance with gray images than with RGB images.	Microsoft Malware Classification dataset (9 families), DumpWare10 (11 families)	DenseNet-121, CNN, CycleGAN
Chaganti et al. (2023) [27]	CNN model for malware classification on Portable Executable (PE) binary files	Multi-view feature fusion-based feature selection approach	• Requires a cautious selection of features Malware binaries only in Windows environments	PE Section, PE Import, PE API, PE Images	SVM, DNN, CNN, LSTM, CNN-LSTM
Panda et al. (2023) [39]	Transfer Learning for Image-Based Malware Detection for IoT	The design of a lightweight system for malware classification which consumes less time and resources;	Difficulty in classifying similar types of malware	Malevis (26 families), MalImg (25 families)	Autoencoder, GRU, MLP
Ali et al. (2020) [23]	A malware detection approach based on N-grams and machine learning	• Feature extraction and representation algorithm Malware behavioral modeling using advance sandbox	Only two classes were considered.	Virushare	Decision Tree, RF, LR, Naive Bayes
Kumar (2021) [34]	Malware classification with fine-tune CNN network model	The traditional and transfer learning approach is used for classification.	Uniform image size as input to the model.	MalImg (25 families), Microsoft Malware Classification dataset (9 families)	ResNet50
Lachtar et al. (2021) [38]	An energy efficient solution based on CNN for mobile malware detection	An approach for using native instructions from mobile apps with CNN	Feature extraction	ARM OAT dataset, x86 OAT dataset	AlexNet, LeNet, InceptionV3

binary code of the malware.

Recently, malware visualization has been used as an alternative and practical approach to malware analysis that does not require deep analysis [12]. Therefore, a new approach was proposed in this paper to improve the efficiency of malware classification in CPS.

This approach converts malicious code into grayscale images. Two pre-trained deep learning models are considered: the AlexNet model and the MobileNet model. Both grayscale malware images and Radon transform-based [15,16] images are used. Experiments are conducted on two large malware datasets [17,18]. The proposed approach classifies malware into families according to common visual characteristics and can be used for subsequent decision-making.

In general, the main contributions of the current study are:

- Development of an ensemble model based on transfer learning that combines malware visualization and radon transform representation.
- The model does not require the development of new features and is based on visual analysis of malware images.
- Experimental datasets of various sizes (Microsoft malware, IoT_Malware, and MalNet-Image datasets) were considered to classify malware in the CPS systems.
- Performance comparison with other well-known machine learning methods has proved the superiority of the proposed approach in terms of efficiency.
- Experimental results show that combining features from two deep neural networks, AlexNet and MobileNet, can effectively classify malicious software in CPS even with small image changes.

This paper is structured as follows. Section 'Related works' describes the literature review. The proposed approach is presented in Section 'Proposed approach'. Section 'Experimental datasets description' describes the considered experimental datasets. The evaluation metrics are shown in Section 'Evaluation metrics'. Section 'Experimental results' presents the experimental results. Section 'Discussion' discusses the advantages and limitations of the proposed approach. Conclusions are given in Section 'Conclusion'.

Related works

In this section, we present related work and discuss the various machine learning methods for malware analysis.

Recently, several studies have been conducted on malware detection and classification in CPS using machine learning (Table 1). When analysing malicious samples, they are converted into text data types or image data to subsequently apply machine learning methods [19,20,21]. In text data, byte encode information in a sequence of letters or numbers. Moreover, in images, a byte represents the intensity of a pixel. In this case, the structure of the binary samples is converted into two-dimensional images, and the features obtained from them are then used to classify malware.

So, Nataraj et al. (2011) proposed a halftone image strategy for malware classification [11]. The malware binary samples are read as vectors of unsigned 8-bit integers and converted into a matrix. It is saved as an image from which the GIST features are extracted [22]. In this approach, classification is based on the k-nearest neighbors (KNN) algorithm using the Euclidean distance metric. This approach can be applied to analyse large datasets and classify packaged and unpackaged binary samples of malicious software.

Ali et al. (2020) developed a method for malware detection based on N-grams and logistic regression (LR) [23]. A dynamic analysis technique was applied to extract an indicator of compromise (IOC) for malicious files. Yan, Zhou, and Zhang (2018) proposed an algorithm of pairwise rotation invariant co-occurrence local binary pattern (PRICoLBP-TFIDF) for malicious code classification with a better-discriminating ability and found that it also has linear separability between different malicious code families [24]. Naeem et al. (2022) developed a new method for malware classification in IoT devices. The essential image features were extracted using a combined local and global feature descriptor (LBP-GLCM) to identify malicious software [25]. Tekerek and Yapici (2022) proposed a data augmentation method based on CycleGAN to improve the accuracy of malware classification [26]. A multi-view feature selection approach was described in [27].

Recent work has focused on applying the transfer learning approach to image-based malware classification [28–32]. Yan et al. (2018) proposed a malware detection method that uses two deep neural networks, convolutional neural network (CNN) and long short-term memory (LSTM) network, followed by a stacking ensemble to fuse them [33]. A deep-learning-based CNN model (MCFT-CNN) that does not require feature engineering was proposed in [34]. Naeem et al. (2023) developed a deep-stacked ensemble model by combining CNNs and a meta-learner (MLP) [35]. Xiao et al. (2021) proposed a malware visualization method that combines Colored Label boxes (CoLab), VGG16, and SVM (Support vector machines). CoLab marks the sections of a PE file to further emphasize the section distribution information in the converted malware image [36]. Carletti et al. (2021) discussed robustness analysis against obfuscation, using ResNet50, InceptionV3, VGG16, and MobileNet for image-based malware classification systems. Mobilenet showed the best result [37]. Lachtar et al. (2021) proposed a solution that harnesses visualization techniques for converting application instructions into images and applied three pre-trained CNN architectures: LeNet, AlexNet, and InceptionV3 [38]. Panda et al. (2023) developed a lightweight system for malware classification based on an autoencoder [39].

Summarizing the above works, we propose an approach for malware classification in CPS based on visual representations using pre-trained deep neural networks (AlexNet and MobileNet). Grayscale byteplots are sent to AlexNet. While the images obtained after the radon transform are used as input images to MobileNet. Next, feature selection is empirically performed. The obtained features are then fed to two fully connected layers for classification following the considered malware families. The proposed approach significantly improves the accuracy of malware classification compared with existing methods.

Proposed approach

This section proposes an approach for malware classification in CPS. The approach includes three main steps: pre-processing, feature extraction, and malware classification. The general scheme of the proposed approach is shown in Fig. 1.

In the first step, the binary sample files from the executable files are converted into an image. The sequence of bytes is converted to binary as a grayscale PNG image using the approach proposed by Nataraj et al. (2011) [11]. It transforms the malware binary to a sequence of 8-bit strings, and each string is converted to a decimal number representing one channel pixel from 0 to 255. Sinusoidal textural features are extracted from the grayscale images using the Radon transform.

Since they do not have a fixed shape, malware images are resized to 224×224 pixels using a bilinear interpolation algorithm while maintaining the image's texture. The samples are then pre-processed and normalized.

In the next step, the Radon transform is applied to the resulting grayscale malware images. Unlike most methods, the Radon transform is a reversible image transformation and can therefore be considered as a method for image texture representation [40].

The algorithm also applies Radon transform to malware images in parallel.

Radon transform is widely used in computed tomography, medicine, geodesy, etc. [15,16]. The two-dimensional Radon transformation has the following form

$$RT(\eta,\mu) = \iint_{R^2} f(x,y)\partial(\eta - x\cos\mu - y\sin\mu)dxdy \tag{1}$$

Here η, μ are polar coordinates.

In this case, the inverse Radon transform can be calculated as

$$f(x,y) = \frac{1}{2p^2}\int_0^{2p}\int_{-\infty}^{\infty}\frac{\frac{\partial}{\partial\eta}RT(\eta,\mu)d\eta d\mu}{x\cos\mu + y\sin\mu - \eta} \tag{2}$$

The result includes sample malware image samples with distinctive textured grayscale features. Additional consideration of images obtained after applying the Radon transform speeds up the interpretation of images to detect malicious programs, highlights the characteristic features of a malware family, and excludes non-essential features.

Figs. 2–4 show grayscale images of malware binaries and corresponding images obtained after applying the Radon transform for the considered datasets. It can be seen that non-packed binary samples belonging to various malware families are very different [41]. However, finding differences in the structure of packaged binary samples of different malware families is only possible using machine learning methods.

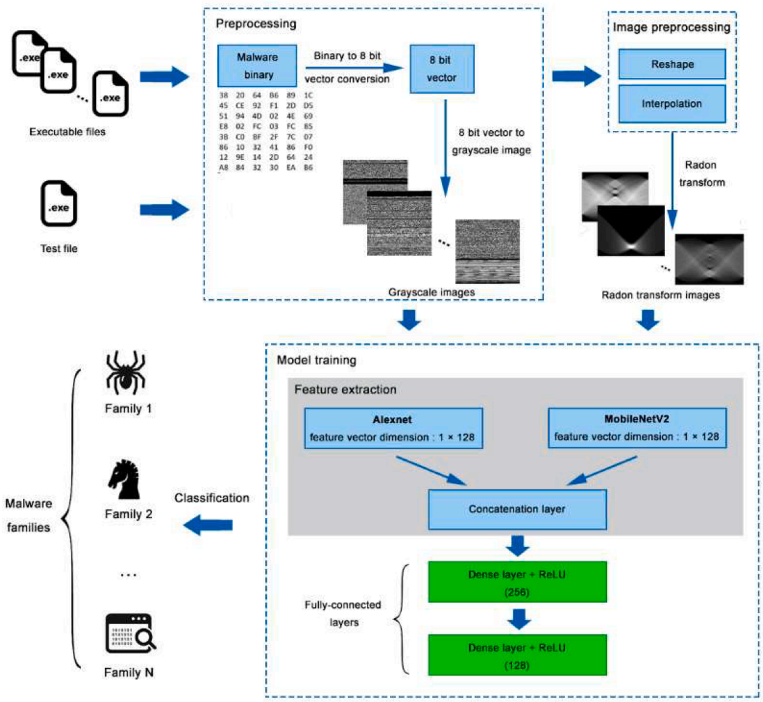


Fig. 1. Flowchart of the proposed approach.

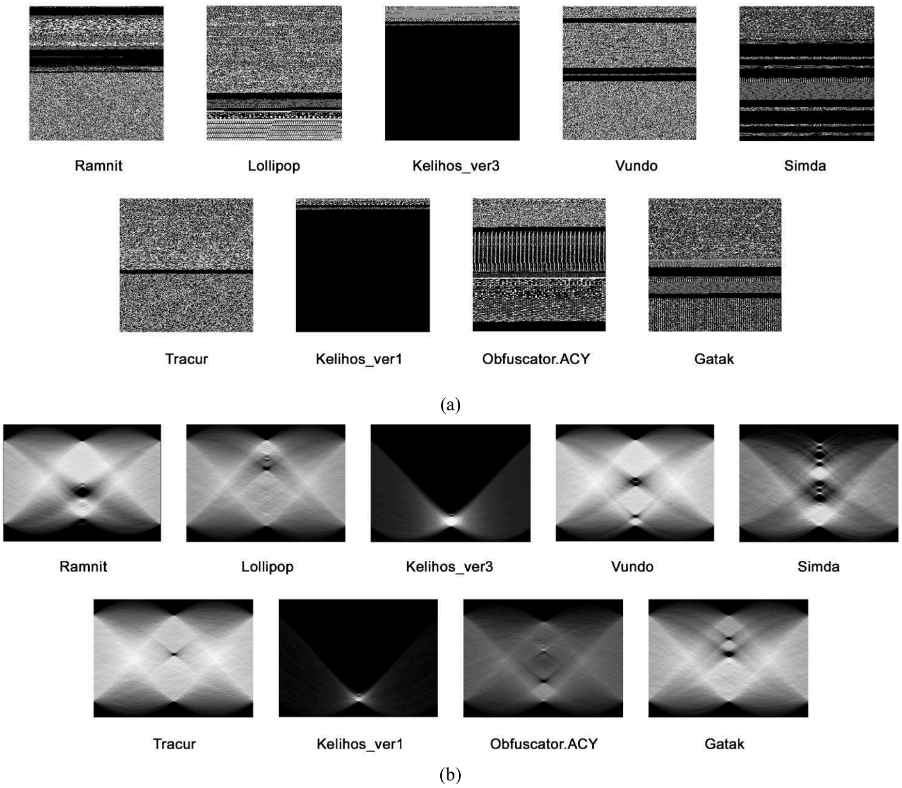


Fig. 2. Examples of malware images (a) and images obtained after applying the Radon transform (b) for Microsoft malware dataset.

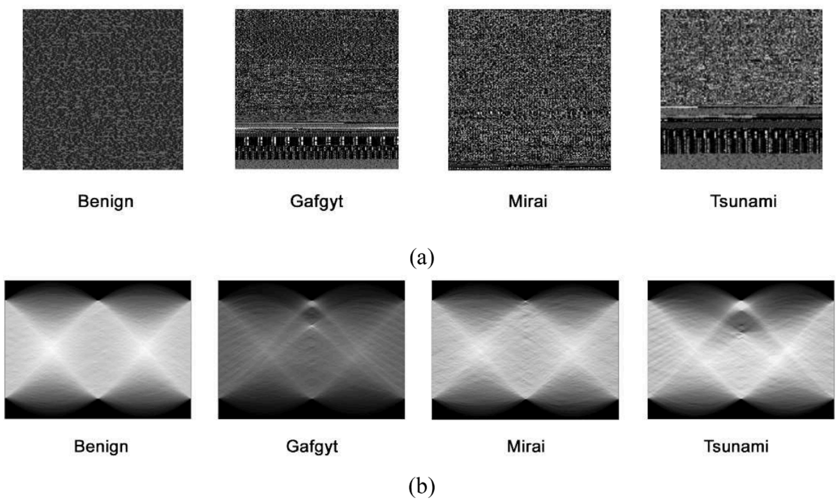


Fig. 3. Examples of malware images (a) and images obtained after applying the Radon transform (b) for IoT_Malware dataset.

The next step involves feature extraction from grayscale images. To achieve this, transfer learning is applied to image datasets. Unlike other machine learning methods, deep learning can quickly and automatically learn features from visual analysis [42]. This study examined two CNN-based models, AlexNet [43] and MobileNetV2 [44], for the feature extraction from the malware images.

Data augmentation is performed because the original dataset is imbalanced. It includes several image transformations: resizing, cropping, rotation, horizontal flip, and others [43].

The CNN model is used because it has excellent performance in image processing. CNN is based on the convolution operation, which reduces the dimension of the feature space and extracts more complex abstract features from an image. It can be described as follows:

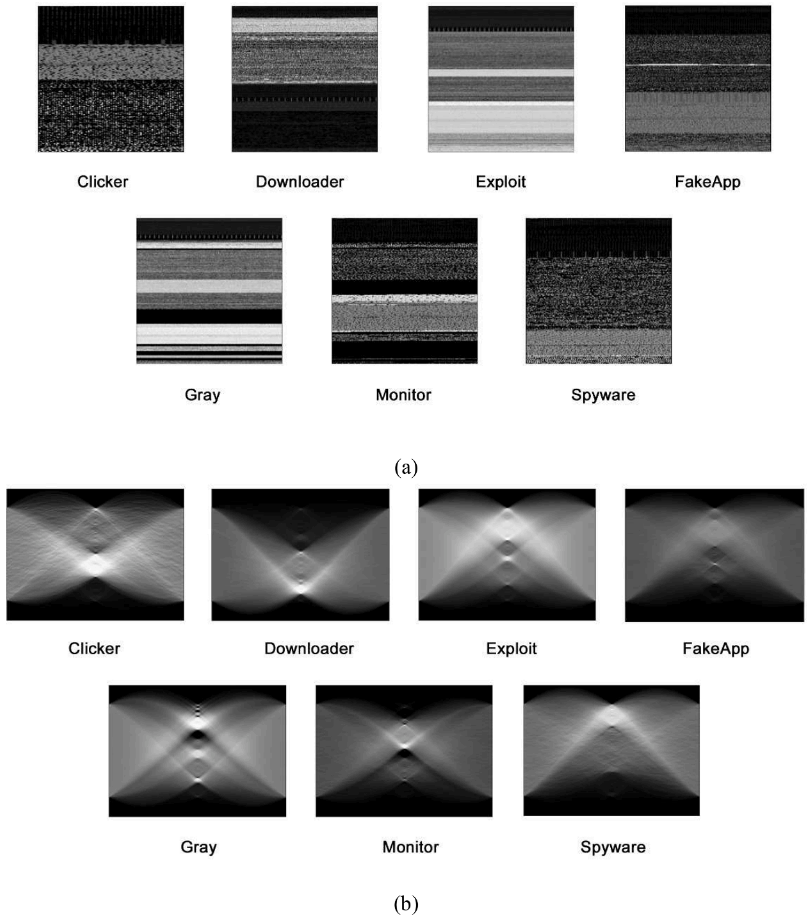


Fig. 4. Examples of malware images (a) and images obtained after applying the Radon transform (b) for MalNet-Image dataset.

$$h(i) = f\left(\sum_{x=1}^{l}\sum_{y=1}^{r} \xi_{xy} \times w_{xy}^{i} + \beta^{i}\right), \tag{3}$$

where $f(.)$ is the activation function (in our case, ReLU (rectified linear unit)), ξ_{xy} is the value of the input data node (x, y) in the filter, and w_{xy}^{i} shows the filter weight, β^{i} is the bias parameter, and l is the length and width of the filter.

AlexNet is a lightweight deep model with strong characterization capabilities and can be easily trained. It trains quickly and has a fast prediction speed. AlexNet has proven its effectiveness and has shown promising results in the following tasks: CPS fault diagnosis [45], network intrusion detection [46], etc. This is why it can be applied to the problem of malware classification. The AlexNet model contains five convolutional layers in the feature extraction module and three fully connected layers in the classification module [43]. AlexNet is difficult to overfit, which is why it is mostly used for identification and classification applications [47].

MobileNet is a small model with low-power parameterized to meet the resource constraints of various cybersecurity applications such as user authentication [24,34,36], fraud detection [48], malware detection [37] and others.

AlexNet and MobileNet are trained on millions of images from the ImageNet database [49]. The model considers the pre-trained weights for the images. Combining the benefits of AlexNet and MobileNet requires more computational resources and increases model run time, but at the same time, improves efficiency in terms of accuracy.

In the proposed model, AlexNet accepts grayscale visual representations of the malware image. Furthermore, MobileNet considers the images obtained after applying the Radon transform. The proposed model replaced the last layers of AlexNet and MobileNet with fully connected layers with 128 neurons each.

Next, the obtained features of the two considered models are concatenated and fed into two fully connected layers, consisting of 256 and 128 neurons for classification by the malware families. Each of the neurons in a fully connected layer is connected to all the neurons of the previous layer and contains a large number of parameters [50].

The hyperparameters of the proposed model based on transfer learning can be seen in Table 2. Adam optimization has shown promising results in image classification and is applied at the training stage, and the softmax activation function is used to calculate the predictive probabilities for all malware families in the output layer.

The predictive probabilities are calculated as follows:

$$p(\hat{x} = j|x) = \frac{\exp(x_{b_i}(j))}{\sum_{s=1}^{S} \exp(x_{b_i}(s))}, \tag{4}$$

where s is the number of malware families.

The performance of the approach is evaluated at the end of each epoch, and the model with the lowest validation error is selected. Table 3 shows the results of hyperparameter tuning for the proposed model.

The pseudo-code of the proposed approach is given below.

Algorithm: Pseudocode for Malware Classification based on transfer learning

Input: Feature vector $FA = \{fa_1, ..., fa_z\}$
Feature vector $FM = \{fm_1, ..., fm_w\}$
Output: Malware family classification $MF = \{mf_1, ..., mf_k\}$
1: Feature vector extraction using AlexNet
2: Feature vector extraction using Mobilenet
3: Feature concatenation
4: repeat
5: for each epoch **do**
6: Calculate Loss function
7: until the convergence condition is met
8: Select the best model
9: Model training
10: Trained model evaluation on test data.
11: return *MF*

The categorical cross-entropy determines the loss for each class [51]. It proves the correct classification of the input data and is calculated as

$$LogLoss = -\frac{1}{N}\sum_{k=1}^{K}\sum_{j=1}^{N} f(j,k)\log(f(j,k)), \tag{5}$$

where N is the number of samples, K is the number of classes, and $f(j,k)$ is the probability that sample j belongs to class k.

Experimental datasets description

The paper considers three datasets to conduct experiments: the Microsoft Malware Classification, IoT_Malware and MalNet-Image datasets.

Microsoft Malware Classification dataset is hosted on Kaggle and available for research [17]. It is 500 gigabytes and contains 9 different malware families, including Rammit, Lollipop, Kelihos_ver3, Vundo, Simda, Tracur, Kelihos_ver1, Obfuscator, and Gatak. Each malware family belongs to one of six types: worms, adware, backdoors, trojans, trojan downloaders, and obfuscated malware (Table 4).

The training set contains 10,868 samples, while the test set contains 10,873. Since the test dataset does not contain malware family labels, only the training dataset is used in this paper. Each data sample has a hexadecimal representation of the binary content of the malware and assembly language source code files generated using the IDA disassembler tool. This information includes build sequences, strings, function calls, etc.

The IoT_Malware dataset for malware detection in CPS was proposed by Alasmary et al. (2020) [18]. Samples were randomly

Table 2
Model parameters.

Parameter	Value
Optimizer type	[Adam, RMSProp, SGD]
Epochs	30
Batch size	[128, 256]
Learning rate	[0.001, 0.0001]
Activation function	ReLU
SVM kernel function	RBF
SVM gamma hyperparameter	0.1
SVM regularization parameter	10
KNN number of neighbors	12
RF number of trees	200
RF number of tree features	5

Table 3
Hyperparameter tuning.

Batch size	Learning rate	Optimizer type	Accuracy (%)		Loss	
			Train	Validation	Train	Validation
128	0.001	Adam	86.26	85.16	0.4477	0.4857
		SGD	91.80	89.96	0.2950	0.2969
		RMSProp	83.98	82.63	0.5744	0.4884
	0.0001	Adam	87.30	87.08	0.4002	0.4071
		SGD	89.96	89.94	0.3033	0.3168
		RMSProp	90.68	87.99	0.2892	0.3787
256	0.001	Adam	92.06	91.25	0.5335	0.5573
		SGD	90.62	88.31	0.3870	0.2846
		RMSProp	86.51	82.46	0.8341	0.9391
	0.0001	Adam	**99.89**	**99.51**	**0.1932**	**0.1887**
		SGD	98.04	97.30	0.3524	0.3690
		RMSProp	92.03	89.53	0.6135	0.5139

Table 4
Microsoft malware dataset description.

	Type	Number of samples	Category	Description
1	Ramnit	1541	Worm	It is capable of rapidly spreading and self-reproducing without human intervention.
2	Lollipop	2478	Adware	Designed for advertising and is a source of income.
3	Kelihos_ver3	2942	Backdoor	Covertly lets an attacker into the system, giving administrator rights.
4	Simda	42		
5	Kelihos_ver1	398		
6	Gatak	1013		
7	Vundo	475	Trojan	It embeds in the device unnoticed by the user and transfers personal data to a third party.
8	Tracur	751	Trojan-Downloader	A Trojan that redirects to another page using Internet search results.
9	Obfuscator.ACY	1228	Obfuscated malware	It is used to bypass anti-virus scanners.

selected from CyberIOCs from January 2018 to late February 2019 [52]. The dataset contains 3,016 benign samples and 13,798 malicious files. The benign and malicious samples in the IoT_Malware dataset were validated using VirusTotal [53]. Malicious software for CPS includes three classes, namely, Gafgyt (11,128 samples), Mirai (2,408 samples), and Tsunami (262 samples) (Table 5).

This work also considered a large publicly available malware dataset, MalNet-Image [54]. It contains over 1.2 million binary images, comprising 47 types and 696 malicious software families. To conduct experiments, we selected 60,437 malware samples and 19,736 benign samples from the dataset (Table 6).

Evaluation metrics

The $N \times N$ confusion matrix is used to calculate four metric types to test the proposed malware classification model. These metrics are accuracy, precision, recall, and F-measure [41].

The considered metrics are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{6}$$

which determines the proportion of correct results obtained by the classifier.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

This metric shows what proportion of objects identified as positive by the classifier is positive.

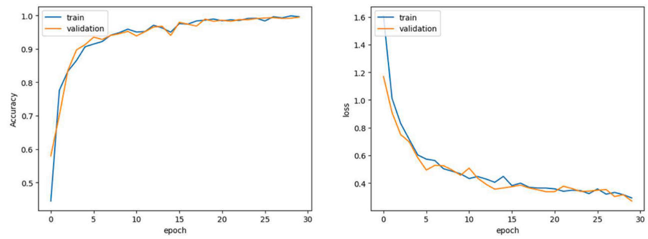
Table 5
IoT_Malware dataset description.

	Type	Number of samples	Category	Description
1	Benign	3,016		
2	Gafgyt	11,128	Backdoor	This malware family consists of ELF files and is used for DDoS attacks.
3	Mirai	2,408	Backdoor	It is malware in Linux that receives commands from control servers to carry out cyberattacks.
4	Tsunami	262	Backdoor	Backdoor.Linux.Tsunami gives the adversary full access to the infected computer, which becomes part of the botnet.

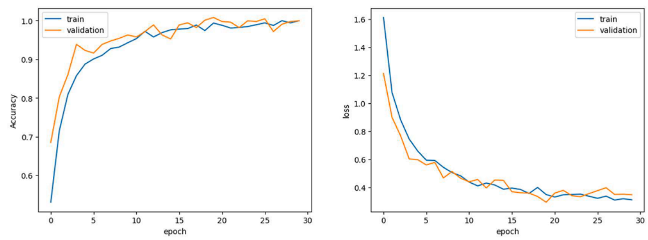
Table 6
MalNet-Image dataset description.

	Type	Number of samples	Category	Description
1	Benign	19,736		
2	Addisplay	17,458	Adware	It is a variant of adware distributed through multiple applications.
3	Adload	533	Adware	It uses a Person-in-The-Middle (PiTM) attack to inject advertisements into web pages.
4	Clicker	591	Adware	It retrieves crawl URL information via Firebase cloud messaging messages.
5	Exploit	5,571	Exploit	It can use a known vulnerability to gain unauthorized access or control of a device.
6	Riskware++SMSsend	547	Riskware	It is an App with an additional module for sending SMS messages that can give more control or access to a device than is allowed.
7	SPR	13,832	Riskware	SPR (Security and Privacy Risk) malware that requires potentially harmful permissions threatening user privacy.
8	Spyware	6,590	Spyware	It can impact the user's privacy, productivity, or control of the computer or device.
9	SMSsend++Trojan	4,524	Trojan	It reaps profit by sending SMS messages to premium-rate numbers.
10	Monitor	1,357	Spyware	It tracks activities on the monitored device and saves information on a remote site.
11	ROG	1,975	Ransomware	It encrypts the personal documents on the victim's computer and then displays a message offering to decrypt the data after payment.
12	Gray	930	Grayware	A software between regular software and a virus belongs to a gray area.
13	Hacktool	542	HackTool	Cybercriminals use it to send a flood of network packets to the targeted machine using brute-forcing and known vulnerabilities.
14	FakeApp	425	Riskware	It replicates the functionalities of other applications to allay suspicions that the App is fake. Some FakeApps are distributed with other legitimate apps.
15	Virus	465	Virus	This malware redirects the user to a web page similar to Office 365, after which an obfuscated malicious JS file will be downloaded.
16	Downloader	4,997	Trojan-Downloader	It secretly downloads malware from a remote server, then installs and executes the files.

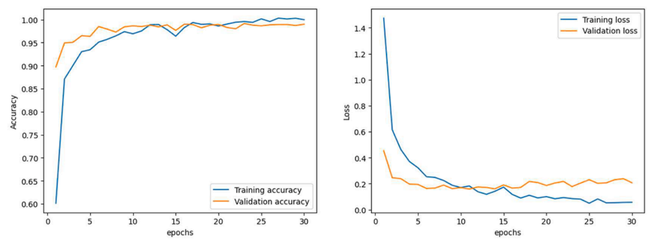
$$Recall = \frac{TP}{TP + FN}, \quad (8)$$



(a) Microsoft malware dataset



(b) IoT_Malware dataset



(c) MalNet-Image dataset

Fig. 5. Malware classification accuracy and loss curves on Microsoft malware dataset (a), IoT_Malware dataset (b), and MalNet-Image dataset (c).

which shows what part of the positive objects was selected by the classifier.

The following metric combines precision and recall metrics:

$$F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{9}$$

Experimental results

In this paper, the proposed approach was implemented in the Python 3.7.12 environment using Keras library with a Tensorflow backend. The experiments were run on a VM with Intel Xeon (R) CPU X5670 @ 2.93 GHz * 24 and 24GB RAM.

To effectively evaluate the performance of the proposed model, the datasets are randomly split into a training (80%) and a validation (20%) set. The approach is run 20 times to reduce the classification error. The experiments are performed on the Microsoft Malware Classification, IoT_Malware, and MalNet-Image datasets.

The resulting grayscale images were normalized. The images were resized to 224×224 and fed to the models' input.

Detection loss curves and accuracy curves for image classification of Microsoft malware dataset, IoT_Malware dataset, and MalNet-Image dataset are shown in Fig. 5. The k-fold cross-validation was used to minimize the overfitting problem.

Experiments showed the optimal value for k equal to 5. There is a steady decrease in verification loss as the number of epochs increases. The proposed model showed a loss of 0.1932 in the training stage and 0.1887 in the validation stage on the Microsoft malware dataset. The classification complexity was observed for the Simda family, which is associated with a small number of samples for this class. At the same time, the accuracy for the Kelijos_ver3 class was 100%.

Tables 7–9 show the performance of the proposed model for various malware families based on precision, recall, and F-measure metrics [55].

According to F-measure Lollipop, Vundo, Kelihos_ver3, Kelihos_ver1, and Obfuscator.ACY classes showed >90% for the Microsoft malware dataset. However, the Ramnit, Tracur, and Gatak malware families were less accurately recognized by the precision metric and accounted for 76%, 82%, and 90%, respectively.

For four classes of the IoT_Malware dataset, the following results were obtained according to the precision metric: Benign (99.50%), Gafgyt (97.99%), Tsunami (96.09%), and Mirai (97.88%). The class of the Mirai malware family aimed at infecting CPS was most accurately identified according to the recall metric. At the same time, the precision and F-measure metrics could identify the Benign software class with almost 100% accuracy.

For the MalNeT-Images dataset, according to the precision, recall, and F-measure metrics, Benign, Addisplay, and SPR were classified correctly. At the same time, the other considered classes of malicious software were also recognized with fairly high accuracy, as shown in Table 9.

Table 10 compares the proposed malware classification approach with various machine-learning methods on various Microsoft malware dataset and IoT_Malware dataset, including MALGRA [23], SVM [24], random forest (RF) [24], KNN [24], and PRICoLBP [24].

Compared with the experimental results of the considered methods, the proposed structure performs well in extracting features from malicious software images using AlexNet and MobileNet deep neural models, as shown in Table 11.

Thus, the above experimental results prove the effectiveness and applicability of the proposed approach to detecting malicious software in CPS.

Discussion

This paper proposes an approach to malware classification in CPS based on the visual similarity of images and feature extraction using AlexNet and MobileNet. This study focuses only on non-packaged binary samples of malicious software.

In this paper, the experiments were conducted on three datasets of malware samples with nine, four and sixteen classes. Tables 10 and 11 compare the proposed approach with various machine learning methods, including RF [56], KNN [24], LSTM [33], SVM [24], and CNN [34,56]. Various studies have been conducted on the application of different feature types, including N-gram, bytecode, and images. Comparison with various deep learning-based models such as VGG16 [36], Xception [57], InceptionV3 [58], and EfficientNetB0 [59] was also performed. Therefore, Hussain et al. (2022) introduced a machine learning-based approach for malware detection in Windows [60]. Despite the high accuracy of the RF method on the Microsoft malware classification dataset, our proposed approach was more accurate and showed a relative improvement of 0.45%. In [38], a CNN-based method for malware classification using images was proposed. It is also based on transfer learning. The limitation of this method is the complexity of feature extraction. It showed lower accuracy compared to our proposed model. Abusnaina et al. (2021) investigated the CPS malware classification task and developed a model that extracts potential malicious behavioral patterns [61]. The proposed ensemble approach based on AlexNet and Mobilenet showed a relative improvement of 2.24% compared to [61]. Thus, the accuracy of the proposed approach outperformed the considered malware classification methods on the Microsoft malware classification and IoT_Malware datasets, as shown in Table 10.

According to Table 11, experiments on a large MalNeT-Images dataset proved the superiority of the proposed approach compared to other deep learning-based methods. Seneviratne et al. [62] proposed a self-supervised model based on the Vision Transformer architecture. Our approach based on AlexNet and Mobilenet showed a relative improvement of 2.27% compared to [62]. In paper [59], a framework based on EfficientNetB0 combined with SVM and RF was developed. Misclassification of some malware samples was observed due to their similarity.

Table 7
Malware classification results of the proposed model on Microsoft malware dataset.

Class	Precision	Recall	F-measure
Ramnit	0.76	0.83	0.80
Lollipop	0.95	0.92	0.94
Kelihos_ver3	**1.00**	**1.00**	**1.00**
Vundo	0.93	0.93	0.93
Simda	0.40	0.29	0.33
Tracur	0.82	0.83	0.83
Kelihos_ver1	0.96	0.95	0.95
Obfuscator.ACY	0.93	0.90	0.92
Gatak	0.90	0.89	0.90

Table 8
Malware classification results of the proposed model on IoT_Malware dataset.

Class	Precision (%)	Recall (%)	F-measure (%)
Benign	**99.50**	99.20	**99.35**
Mirai	97.88	**99.57**	98.72
Tsunami	96.09	92.81	94.42
Gafgyt	97.99	98.23	98.11

Table 9
Malware classification results of the proposed model on MalNet-Image dataset.

Malware	Precision (%)	Recall (%)	F-measure (%)
Benign	**100**	**100**	**100**
Addisplay	**100**	**100**	**100**
Adload	98.12	92.86	95.42
Clicker	97.78	**100**	98.88
Exploit	**100**	98.18	99.08
Riskware++SMSsend	92.00	100	95.83
SPR	**100**	**100**	**100**
Spyware	97.13	96.98	97.05
SMSsend++Trojan	**100**	98.88	99.44
Monitor	97.72	98.17	97.94
ROG	99.01	**100**	99.50
Gray	99.20	**100**	99.60
Hacktool	99.80	89.74	94.50
FakeApp	**100**	96.55	98.25
Virus	95.73	96.88	96.30
Downloader	99.32	**100**	99.66

Table 10
Performance comparison of the proposed approach with various machine learning methods.

Methods	Feature type	Dataset	Accuracy (%)
LR [23]	N-gram	Microsoft malware dataset	98.40
LBP, KNN, SVM, RF, Gradient boost classifier [24]	PRICoLBP		98.60
RF [56]	bytecode		99.44
Proposed approach (AlexNet)	images		90.23
Proposed approach (Mobilenet)	images		89.52
Proposed approach (AlexNet+Mobilenet)	images		**99.89**
RF	N-gram	IoT_Malware dataset	98.46
SVM	N-gram		97.53
KNN	N-gram		95.02
Proposed approach (AlexNet)	images		96.71
Proposed approach (Mobilenet)	images		90.28
Proposed approach (AlexNet+Mobilenet)	images		**99.95**

A limitation of the study is that it did not analyse RGB images but only grayscale images of malware samples. This task will be explored in the future. Since unpacked binary samples were considered in the work, it is planned to develop a method for detecting the obfuscation of packaged malware files. Therefore, O'Shaughnessy & Sheridan (2022) considered Shannon entropy [63] to determine the level of obfuscation [64]. It is also planned to develop models for detecting and quickly responding to zero-day attacks.

Table 11
Performance evaluation of the proposed approach with state-of-the-art methods based on deep learning.

References	Models	Dataset	Accuracy (%)
Xiao et al. (2021) [35]	VGG16	Microsoft malware dataset	98.94
Carletti et al. (2021) [36]	MobileNet		99.25
Lo et al. (2019) [57]	Xception		99.17
Lachtar et al. (2021) [38]	LeNet, AlexNet, InceptionV3		99.70
Kumar (2021) [34]	CNN		98.64
Ahmed et al. (2023) [58]	InceptionV3		99.60
Proposed approach	AlexNet+Mobilenet		**99.89**
Abusnaina et al. (2021) [61]	Adversarial learning	IoT_Malware dataset	97.67
Belguendouz et al. (2022) [56]	CNN		95.00
Proposed approach	AlexNet+Mobilenet		**99.95**
Seneviratne et al. (2022) [62]	Vision Transformer	MalNet-Image dataset	97.00
Yadav et al. (2022) [59]	EfficientNetB0+SVM+RF		92.90
Proposed approach	AlexNet+Mobilenet		**99.20**

Conclusion

In this paper, an approach for CPS malware classification based on pre-trained deep neural networks was proposed. Experiments were conducted on datasets of different sizes (Microsoft malware, IoT_Malware, and MalNeT-Images datasets) to validate the effectiveness of the proposed model. We demonstrated the ability to apply Radon transform to grayscale images in the proposed model to improve the classification accuracy of new malware binaries. Two different models based on Alexnet and MobileNet were used in the ensemble model to classify malware, which made it possible to effectively recognize malware families such as Gafgyt, Tsunami, Mirai, and Kelihos for the IoT_Malware dataset. The Addisplay, Adload, Exploit, Riskware++SMSsend, Spyware, SMSsend++Trojan, ROG, and Downloader classes were also recognized with high accuracy for a large MalNet-Image dataset. The model demonstrated consistent performance with 99.89%, 99.95%, and 99.20% accuracy on the Microsoft malware dataset, IoT_Malware dataset, and MalNeT-Images dataset, respectively.

Thus, given the scale of CPS cybersecurity threats and the large number of unprotected devices, this research is an essential step toward developing new systems protection methods and practical tools. This work can be useful to researchers and practitioners for the timely detection and elimination of CPS threats. Eq. (1)-(9)

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This work was supported by the Science Foundation of the State Oil Company of Azerbaijan Republic (SOCAR) (Contract No. 3LR-AMEA).

References

[1] Yang B, Yu Z, Cai Y. Malicious software spread modeling and control in cyber-physical systems. Knowl-Based Syst 2022;248:108913. https://doi.org/10.1016/j.knosys.2022.108913.
[2] Piqueira JRC, Cabrera MA, Batistela CM. Malware propagation in clustered computer networks. Physica A 2021;573:125958. https://doi.org/10.1016/j.physa.2021.125958.
[3] Yu Z, Gao H, Wang D, Alnuaim AA, Firdausi M, Mostafa AM. SEI2RS malware propagation model considering two infection rates in cyber–physical systems. Physica A 2022;597:127207. https://doi.org/10.1016/j.physa.2022.127207.
[4] Humayun M, Niazi M, Jhanjhi N, Alshayeb M, Mahmood S. Cyber security threats and vulnerabilities: a systematic mapping study, Arab. J Sci Eng 2020;45:3171–89. https://doi.org/10.1007/s13369-019-04319-2.
[5] Pivoto DG, F.de Almeida L, da Rosa Righi R, Rodrigues JJ, Lugli AB, Alberti AM. Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: a literature review. J Manufact Syst 2021;58:176–92. https://doi.org/10.1016/j.jmsy.2020.11.017.
[6] Krotofil M, Cárdenas AA, Manning B, Larsen J. CPS: driving cyber-physical systems to unsafe operating conditions by timing DoS attacks on sensor signals. In: Proceedings of the 30th Annual Computer Security Applications Conference; 2014. p. 146–55. https://doi.org/10.1145/2664243.2664290.
[7] Ngo Q, Nguyen H, Le V, Nguyen D. A survey of IoT malware and detection methods based on static features. ICT Express 2020;6(4):280–6. https://doi.org/10.1016/j.icte.2020.04.005.
[8] Kayan H, Nunes M, Rana O, Burnap P, Perera C. Cybersecurity of industrial cyber-physical systems: a review. ACM Comput Surv 2022;54(11s):1–35. https://doi.org/10.1145/3510410.

[9] Naeem H, Ullah F, Naeem MR, Khalid S, Vasan D, Jabbar S, Saeed S. Malware detection in industrial internet of things based on hybrid image visualization and deep learning model. Ad Hoc Netw 2020;105:102154. https://doi.org/10.1016/j.adhoc.2020.102154.
[10] Damodaran A, Troia FD, Visaggio CA, Austin TH, Stamp M. A comparison of static, dynamic, and hybrid analysis for malware detection. J Comput Virol Hack Tech 2017;13:1–12. https://doi.org/10.1007/s11416-015-0261-z.
[11] Nataraj L, Yegneswaran V, Porras P, Zhang J. A comparative assessment of malware classification using binary texture analysis and dynamic analysis. In: Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence; 2011. p. 21–30. https://doi.org/10.1145/2046684.2046689.
[12] Roseline SA, Geetha S, Kadry S, Nam Y. Intelligent vision-based malware detection and classification using deep random forest paradigm. IEEE Access 2020;8: 206303–24. https://doi.org/10.1109/ACCESS.2020.3036491.
[13] Zhong F, Chen Z, Xu M, Zhang G, Yu D, Cheng X. Malware-on-the-Brain: illuminating malware byte codes with images for malware classification. IEEE Trans Comput 2022;72(2):438–51. https://doi.org/10.1109/TC.2022.3160357.
[14] Shankarapani MK, Ramamoorthy S, Movva RS, Mukkamala S. Malware detection using assembly and API call sequences. J Comput Virol 2011;2(7):107–19. https://doi.org/10.1007/s11416-010-0141-5.
[15] Ziou D, Nacereddine N, Goumeidane AB. Scale space Radon transform. IET Image Process 2021;15(9):2097–111. https://doi.org/10.1049/ipr2.12180.
[16] Jin S, Yin J, Tian M, Feng S, Thompson SG, Li Z. Practical speed measurement for an intelligent vehicle based on double radon transform in urban traffic scenarios. Meas Sci Technol 2020;32(2):025114. https://doi.org/10.1088/1361-6501/abb5d9.
[17] Microsoft BIG. Kaggle: Microsoft malware classification challenge. 2015. URL, https://www.kaggle.com/c/malware-classification.
[18] Alasmary H, Abusnaina A, Jang R, Abuhamad M, Anwar A, Nyang D, Mohaisen D. Soteria: detecting adversarial examples in control flow graph-based malware classifiers. In: Proceedings of the IEEE 40th International Conference on Distributed Computing Systems (ICDCS); 2020. p. 888–98. https://doi.org/10.1109/icdcs47774.2020.00089.
[19] Kadri MA, Nassar M, Safa H. Transfer learning for malware multi-classification. In: Proceedings of the 23rd ACM International Database Applications & Engineering Symposium; 2019. p. 1–7. https://doi.org/10.1145/3331076.3331111.
[20] Vasan D, Alazab M, Wassan S, Naeem H, Safaei B, Zheng Q. IMCFN: image-based malware classification using fine-tuned convolutional neural network architecture. Comput Netw 2020;171:107138. https://doi.org/10.1016/j.comnet.2020.107138.
[21] Shu L, Dong S, Su H, Huang J. Android malware detection methods based on convolutional neural network: a survey. IEEE Trans Emerg Topics Comput Intell 2023;7(5):1330–50. https://doi.org/10.1109/TETCI.2023.3281833.
[22] Olivia A, Torralba A. Modeling the shape of a scene: a holistic representation of the spatial envelope. Int J Comput Vision 2001;42(3):145–75. https://doi.org/10.1023/A:1011139631724.
[23] Ali M, Shiaeles S, Bendiab G, Ghita B. MALGRA: machine learning and n-gram malware feature extraction and detection system. Electron (Basel) 2020;9(11): 1777. https://doi.org/10.3390/electronics9111777.
[24] Yan H, Zhou H, Zhang H. Automatic malware classification via PRICoLBP. Chin J Electron 2018;27(4):852–9. https://doi.org/10.1049/cje.2018.05.001.
[25] Naeem H, Cheng X, Ullah F, Jabbar S, Dong S. A deep convolutional neural network stacked ensemble for malware threat classification in internet of things. J Circuits Syst Comput 2022;31(17):2250302. https://doi.org/10.1142/S0218126622503029.
[26] Tekerek A, Yapici MM. A novel malware classification and augmentation model based on convolutional neural network. Comput Sec 2022;112:102515. https://doi.org/10.1016/j.cose.2021.102515.
[27] Chaganti R, Ravi V, Pham TD. A multi-view feature fusion approach for effective malware classification using Deep Learning. J Inform Sec Applic 2023;72: 103402. https://doi.org/10.1016/j.jisa.2022.103402.
[28] Nguyen H, Di Troia F, Ishigaki G, Stamp M. Generative adversarial networks and image-based malware classification. J Comput Virol Hack Tech 2023:1–17. https://doi.org/10.1007/s11416-023-00465-2.
[29] Bhodia N, Prajapati P, Di Troia F, Stamp M. Transfer learning for image-based malware classification. In: Proceedings of the 5th International Conference on Information Systems Security and Privacy (ICISSP); 2019. p. 719–26. https://doi.org/10.5220/0007701407190726.
[30] Prajapati P, Stamp M. An empirical analysis of image-based learning techniques for malware classification. In: Stamp M, Alazab M, Shalaginov A, editors. Malware analysis using artificial intelligence and deep learning. Cham: Springer; 2019. https://doi.org/10.1007/978-3-030-62582-5_16.
[31] Kim S. PE header analysis for malware detection. master's thesis. San Jose State University; 2018. URL, https://scholarworks.sjsu.edu/etd_projects/624/.
[32] Yajamanam S, Selvin VRS, Di Troia F, Stamp M. Deep learning versus gist descriptors for image-based malware classification. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP); 2018. p. 553–61. https://doi.org/10.5220/0006685805530561.
[33] Yan J, Qi Y, Rao Q. Detecting malware with an ensemble method based on deep neural network. Secur Commun Netw 2018;2018:1–16. https://doi.org/10.1155/2018/7247095.
[34] Kumar S. MCFT-CNN: malware classification with fine-tune convolution neural networks using traditional and transfer learning in Internet of Things. Future Gener Comput Syst 2021;125:334–51. https://doi.org/10.1016/j.future.2021.06.029.
[35] Naeem H, Dong S, Falana OJ, Ullah F. Development of a deep stacked ensemble with process based volatile memory forensics for platform independent malware detection and classification. Expert Sys Appl 2023;223:119952. https://doi.org/10.1016/j.eswa.2023.119952.
[36] Xiao M, Guo C, Shen G, Cui Y, Jiang C. Image-based malware classification using section distribution information. Comput Secur 2021;110:102420. https://doi.org/10.1016/j.cose.2021.102420.
[37] Carletti V, Greco A, Saggese A, Vento M. Robustness evaluation of convolutional neural networks for malware classification. In: Proceedings of the Italian Conference on Cybersecurity (ITASEC); 2021. p. 414–23. URL, https://ceur-ws.org/Vol-2940/paper35.pdf.
[38] Lachtar N, Ibdah D, Bacha A. Toward mobile malware detection through convolutional neural networks. IEEE Embedded Syst Lett 2021;13(3):134–7. https://doi.org/10.1109/les.2020.3035875.
[39] Panda P, Kumar C U O, Marappan S, Ma S, Manimurugan S, Veesani Nandi D. Transfer learning for image-based malware detection for IoT. Sensors 2023;23(6): 3253. https://doi.org/10.3390/s23063253.
[40] Shifat-E-Rabbi M, Yin X, Rubaiyat AHM, Li S, Kolouri S, Aldroubi A, Nichols JM, Rohde GK. Radon cumulative distribution transform subspace modeling for image classification. J Math Imaging Vis 2021;63:1185–203. https://doi.org/10.1007/s10851-021-01052-0.
[41] Cui Z, Xue F, Cai X, Cao Y, Wang G, Chen J. Detection of malicious code variants based on deep learning. IEEE Trans Ind Informatics 2018;14(7):3187–96. https://doi.org/10.1109/TII.2018.2822680.
[42] Zhang Z, Zhang X. MIDCAN: a multiple input deep convolutional attention network for COVID-19 diagnosis based on chest CT and chest X-ray. Pattern Recognit Lett 2021;150:8–16. https://doi.org/10.1016/j.patrec.2021.06.021.
[43] Krizhevsky A, Sutskever I, Hinton G. Imagenet classification with deep convolutional networks. Commun ACM 2012;60(6):84–90. https://doi.org/10.1145/3065386.
[44] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. MobileNets: efficient convolutional neural networks for mobile vision applications. 2017. arXiv preprint arXiv:1704.04861.
[45] Li G, Hu J, Shan D, Ao J, Huang B, Huang Z. A CNN model based on innovative expansion operation improving the fault diagnosis accuracy of drilling pump fluid end. Mech Syst Signal Process 2023;187:109974. https://doi.org/10.1016/j.ymssp.2022.109974.
[46] Dong Y, Wang R, He J. Real-time network intrusion detection system based on deep learning. In: Proceedings of IEEE 10th International Conference on Software Engineering and Service Science (ICSESS); 2019. p. 1–4. https://doi.org/10.1109/ICSESS47205.2019.9040718.
[47] Li B. Hearing loss classification via AlexNet and extreme learning machine. Int J Cogn Comput Eng 2021;2:144–53. https://doi.org/10.1016/j.ijcce.2021.09.002.
[48] Khan RU, Zhang X, Kumar R. Analysis of ResNet and GoogleNet models for malware detection. J Comput Virol Hack Tech 2019;15:29–37. https://doi.org/10.1007/s11416-018-0324-z.
[49] Burnaev E, Smolyakov D. One-class SVM with privileged information and its application to malware detection. In: Proceedings of IEEE 16th International Conference on Data Mining Workshops (ICDMW); 2016. p. 1–8. https://doi.org/10.1109/ICDMW.2016.0046.

[50] Ahmed MU, Kim YH, Rhee PK. EER-ASSL: combining rollback learning and deep learning for rapid adaptive object detection. KSII Trans Internet Inf Syst 2020; 14:4776–94. https://doi.org/10.3837/tiis.2020.12.009.
[51] Kingma DP, Ba J. Adam: a method for stochastic optimization. 2014. arXiv preprint arXiv:1412.6980.
[52] Cyberiocs, 2023. https://freeiocs.cyberiocs.pro/.
[53] VirusTotal, 2023. https://www.virustotal.com.
[54] Freitas S, Duggal R, Chau DH. Malnet: a large-scale cybersecurity image database of malicious software. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM); 2022. p. 3948–52. https://doi.org/10.1145/3511808.3557533.
[55] Ferri C, Hernández-Orallo J, Modroiu R. An experimental comparison of performance measures for classification. Pattern Recognit Lett 2009;30(1):27–38. https://doi.org/10.1016/j.patrec.2008.08.010.
[56] Belguendouz H, Guerid H, Kaddour M. Static classification of IoT malware using grayscale image representation and lightweight convolutional neural networks. In: Proceedings of the IEEE 5th International Conference on Advanced Communication Technologies and Networking (CommNet); 2022. p. 1–8. https://doi.org/10.1109/CommNet56067.2022.9993956.
[57] Lo WW, Yang X, Wang Y. An Xception convolutional neural network for malware classification with transfer learning. In: Proceedings of the 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS); 2019. p. 1–5. https://doi.org/10.1109/ntms.2019.8763852.
[58] Ahmed M, Afreen N, Ahmed M, Sameer M, Ahamed J. An inception V3 approach for malware classification using machine learning and transfer learning. Int J Intell Netw 2023;4:11–8. https://doi.org/10.1016/j.ijin.2022.11.005.
[59] Yadav P, Menon N, Ravi V, Vishvanathan S, Pham TD. A two-stage deep learning framework for image-based android malware detection and variant classification. Comput Intell 2022;38(5):1748–71. https://doi.org/10.1111/coin.12532.
[60] Hussain A, Asif M, Ahmad MB, Mahmood T, Raza MA. Malware Detection Using Machine Learning Algorithms for Windows Platform. In: Ullah A, Anwar S, Rocha Á, Gill S, editors. Proceedings of International Conference on Information Technology and Applications. Lecture Notes in Networks and Systems. 350. Singapore: Springer; 2022. https://doi.org/10.1007/978-981-16-7618-5_53.
[61] Abusnaina A, Abuhamad M, Alasmary H, Anwar A, Jang R, Salem S, Nyang D, Mohaisen D. DL-FHMC: deep learning-based fine-grained hierarchical learning approach for robust malware classification. IEEE Trans Dependable Secur Comput 2021;19(5):3432–47. https://doi.org/10.1109/TDSC.2021.3097296.
[62] Seneviratne S, Shariffdeen R, Rasnayaka S, Kasthuriarachchi N. Self-supervised vision transformers for malware detection. IEEE Access 2022;10:103121–35. https://doi.org/10.1109/ACCESS.2022.3206445.
[63] Shannon CE. A mathematical theory of communication. Bell Syst Tech J 1948;27(3):379–423. https://doi.org/10.1145/584091.584093.
[64] O'Shaughnessy S, Sheridan S. Image-based malware classification hybrid framework based on space-filling curves. Comput Secur 2022;116:102660. https://doi.org/10.1016/j.cose.2022.102660.