IOT-MDEDTL: IoT Malware Detection based on Ensemble Deep Transfer Learning

Q. Kh. Kadhim[1*], A. Q. A. Sharhan Al-Sudani[2], I. Amjed Almani[3], T. Alghazali[4], H. khalid Dabis[5], A. Taha Mohammed[6], S. G. Talib[7], R. A. Mahmood[8], Z. Tariq Sahi[9], and Y. S. Mezaal[10]
1- English Language Department, Al-Mustaqbal University College, Babylon, Iraq.
Email: qasim.khlaif@mustaqbal-college.edu.iq (Corresponding author)
2- Al-Manara College For Medical Sciences, Maysan, Iraq.
Email: ahmedqasim@uomanara.edu.iq
3- Department of Computer Technology Engineering, Al-Hadba University College, Iraq.
Email: enas.am@hcu.edu.iq
4- College of Media, Department of Journalism, The Islamic University in Najaf, Najaf, Iraq.
Email: gazali.tawfeeq@gmail.com
5- College of Islamic Science, Ahl Al Bayt University, Kerbala, Iraq.
Email: m.hasan7379@gmail.com
6- The University of Mashreq, Iraq.
Email: atheer.taha@uom.edu.iq
7- Law Department, Al-Mustaqbal University College, Babylon, Iraq.
Email: saadghazi@mustaqbal-college.edu.iq
8- Medical device engineering, Ashur University College, Baghdad, Iraq.
Email: rawnaq.adnan@au.adu.iq
9- Department of Dentistry, Al-Zahrawi University College, Karbala, Iraq.
Email: zahraatariq@g.alzahu.edu.iq
10- Al-Esraa University College, Baghdad, Iraq.
Email: yaqeen@esraa.edu.iq

Received: March 2022 Revised: May 2022 Accepted: June 2022

ABSTRACT:
The internet of Things (IoT) is a promising expansion of the traditional Internet, which provides the foundation for millions of devices to interact with each other. IoT enables these smart devices, such as home appliances, different types of vehicles, sensor controllers, and security cameras, to share information, and this has been successfully done to enhance the quality of user experience. IoT-based mediums in day-to-day life are, in fact, minuscule computational resources, which are adjusted to be thoroughly domain-specific. As a result, monitoring and detecting various attacks on these devices becomes feasible. As the statistics prove, in the Mirai and Brickerbot botnets, Distributed Denial-of-Service (DDoS) attacks have become increasingly ubiquitous. To ameliorate this, in this paper, we propose a novel approach for detecting IoT malware from the preprocessed binary data using transfer learning. Our method comprises two feature extractors, named ResNet101 and VGG16, which learn to classify input data as malicious and non-malicious. The input data is built from preprocessing and converting the binary format of data into gray-scale images. The feature maps obtained from these two models are fused together to further be classified. Extensive experiments exhibit the efficiency of the proposed approach in a well-known dataset, achieving the accuracy, precision, and recall of 96.31%, 95.31%, and 94.80%, respectively.

KEYWORDS: Internet of Things, Deep Learning, Ensemble Learning, Transfer Learning, Convolutional Neural Networks, Malware Detection.

1. INTRODUCTION

One of the most influential inventions in the history of computer science is the creation of the Internet [1]. It is the new driving force of all technologies which exists today, no economic and industrial interrelations between countries around the globe can be done without the Internet [2]. The evolution of the Internet has caused a new phenomenon to be born, which is called The Internet of Things (IoT) [3]. IoT has been considered the future of the Internet, by which a large

Paper type: Research paper
DOI: https:// 10.30486/mjee.2022.696506
How to cite this paper: Q. Kh. Kadhim, A. Q. A. Sharhan Al-Sudani, I. A. Almani, T. Alghazali, H. kh. Dabis, A. T. Mohammed, S. G. Talib, R. A. Mahmood, Z. T. Sahi, and Y. S. Mezaal, "**IOT-MDEDTL: IoT Malware Detection based on Ensemble Deep Transfer Learning"**, Majlesi Journal of Electrical Engineering, Vol. 16, No. 3, pp. 47-54, 2022.

number of devices can be connected to each other. These smart devices can interact with each other and share information, and this provides the end-users with a better experience [4]. The applications of IoT are manifold, ranging from healthcare systems [5], automotive, industrial applications [6], transportation [7], etc. Before IoT, these devices could only function on domain-specific tasks, which were assigned by the pre-defined rules. In stark contrast to this, IoT provides a Central Processing Unit (CPU)-based environment for these mediums, and this can make them more intelligent and give them more computational power [8].

Although IoT has the advantages mentioned above, the smartness, which comes with IoT, can open new opportunities for adversarial attacks [9]. These vulnerabilities have drawn considerable attention from the researchers since the potential threatening attacks can be a danger for the devices themselves and the users who harness these kinds of facilities [10]. It is worth mentioning that today's IoT-based systems are far from being thoroughly secured and the main reason for this is the lack of a set of unified and standard principles for securing various types of hardware and software platforms [11]. Additionally, compared with machines such as personal computers and laptops, the shortage of computational resources on these smart devices prevents us from adopting common policies normally implemented to make such systems robust to attackers [12]. Nonetheless, cloud-based systems can give us the opportunity to develop protective modules to detect and fight against cyber-attacks. One of the biggest examples of such attacks is malware which is intentionally made to intrusively enter the victim's machines and steal data or damage the systems. Examples of malware are viruses, worms, spyware, adware, and trojans [13].

So far, a variety of machine learning and deep learning-based models have been introduced in the literature for detecting and classifying malicious and non-malicious malware. In [14], a large-scale system in which random projection and a neural network are proposed for the classification task is proposed. Their results show a minimum of less than 1.00% error rate with a 16000 dimension for the random projections. Firdausi et al. [15] have analyzed the behavior of malware on an emulated environment by extracting reports from these behaviors. Then these reports are altered into sparse vectors and fed to different machine learning-based models for the downstream classification task. Their best result belongs to the Decision Tree classifier with an accuracy of 96.8%, recall of 95.9%, and false-positive rate of 2.4%. In [16], a novel lightweight method is proposed for detecting IoT botnets. They used a feature extraction process for function-call graphs, called PSI-Graph, and these are made for each executable file. They achieved an accuracy of 98.7% in a dataset containing 11,200 ELF files. Alasmary et al., in [17], have designed a malware detection system that utilized Control Flow Graphs (CFGs). In their study, it was demonstrated that IoT malware samples contain a big number of edges with a small number of nodes, and this demonstrates a more resourceful structure with higher complexity. Their achieved results show an accuracy of 99.66%. Although the performance of the previous works has been remarkably good, the fact that malware images can be a reliable way of detecting intrusive actions in IoT-based systems is not investigated. Further, the use of transfer learning which makes the training phase of the CNN-based models is examined, and this can be critical where the software and hardware resources are in shortage.

In this paper, in order to address the disadvantages of the previous works, we propose a framework for detecting malicious and intrusive malware existing in the IoT-based models. Our approach functions on the images obtained via a preprocessing step on binary malware data. Two CNN-based models are utilized in order to generate feature maps suitable for binary classification of the data into benign and malicious classes. The main contributions of this study are itemized in the following:

1. We propose a framework for malicious malware in IoT-based models.
2. We apply the transfer learning technique, which makes the training process of the proposed approach better and easiest to converge.
3. The dataset used in this study is real-world data gathered from real IoT-based environments, and this makes our evaluations more reliable.

2. TECHNICAL WORK PREPARATION

2.1. Overview

In this section, we include our proposed methodology. Fig. 1 illustrates an overview of our approach for feature extraction and classification of the data, which is reformatted to act as an image for the CNN-based networks.

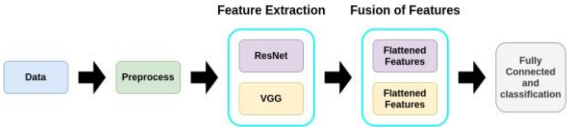


Fig. 1. The overview of the proposed methodology.

As seen in Fig. 1, the data is first pre-processed to be prepared for the feature extraction module. This module contains two models, each of which generates feature maps. Following this, these features are flattened and fused together to be fed to the fully connected classification head.

2.2. Malware Image Classification

The original format of the data used in this study is the format of a malware binary. However, one of the most effective ways of extracting patterns in this type of data is the conversion to an 8-bit sequence [18]. Then, these can be reformatted to become similar to an image with only one channel. These images, which are gray-scale, can then be fed to CNN-based feature extraction modules with the final goal of designing a decision-making system that predicts the danger of intruders.

2.3. Preprocessing

As stated in section 2.2, to prepare our dataset for the proposed algorithm, we need to preprocess our data. This process is shown in detail in Fig. 2.

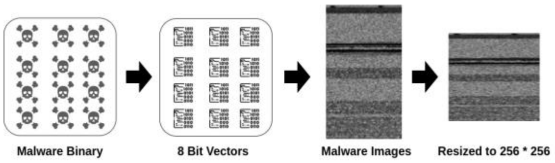


Fig. 2. Different steps of the preprocessing stage in the proposed method.

Further, in order for the proposed model to train better, we normalize the images to have a specific and fine-tuned standard deviation and mean. These values are shown in Table X and are default values for the training ResNet in its base paper.

2.4. Convolutional Neural Networks And Feature Extraction

Convolutional Neural Networks (CNNs), introduced in [19], have one most used types of neural networks in computer vision [20]. Their performance in many challenging tasks such as face recognition [21], object detection [22], image classification [23], image restoration [24], image captioning [25], industrial applications [26], etc., is remarkably good and this makes them exclusively fruitful in terms of image analysis. Their architecture comprises thousands to millions of neurons, where each neuron contributes to the learning phase of the whole network by learning a set of weights and biases [27]. These neurons reside in layers within the network and are called hidden layers [28]. The hidden layers in CNN-based models comprise convolutional layers, pooling layers, normalization layers, and fully connected layers. The core of the convolutional layers is the convolution operation, which is applied by sliding a kernel over the input data to do an element-wise multiplication [29]. The pooling layers are used with the goal of dimensionality reduction and can be done by taking a single output for a cluster of neurons. Moreover, in these models, the convolutional part of the model is famously called the backbone, which is used for feature extraction purposes, and the fully connected ones are used for tasks such as classification.

Heretofore, several CNN-based architectures have been introduced in the literature. Two of the most known networks are ResNet [30] and VGG [31], each of which includes multiple networks with different depths and a number of trainable parameters. In the following, their structure is elaborated.

ResNet refers to a family of deep neural architectures, which initially came with the object of addressing the issue of vanishing gradient issue in deep networks [32]. According to the universal approximation theorem, it is theoretically accepted that a feedforward network containing even a single layer is able to represent any function [33]. However, this is not feasible since the layer should be extremely large, and the network becomes prone to overfitting [34]. This conundrum has motivated the research community to opt for making the networks deeper with more hidden layers. This is where the gradient becomes extremely small when it reaches the initial layers in the process of backpropagation, and it prohibits us from making the designed neural networks more and more deep [35]. ResNet addresses this issue by introducing shortcut connections [36]. These connections skip several layers on their way and, in fact, are integrated with the output of the skipped layers, followed by the next layers [37]. This means that given x as the input of a layer, its output can be defined as equation 1.

$$F(x) = G(wx + b) \tag{1}$$

Where G denotes a non-linear activation function. However, when we deal with residual blocks within ResNet models, equation 1 changes to equation 2, where the input x, is added to $F(x)$. Equation 2 demonstrates this addition.

$$F(x) = G(wx + b) + x \tag{2}$$

In this paper, we chose ResNet101. Its architecture is illustrated in Fig. 3. As is observed in Fig. 3, ResNet 101 has four convolutional blocks with skip connections between each of them. Immediately before and after these blocks, pooling layers exist in the model.

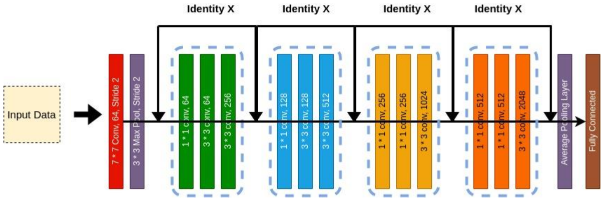


Fig. 3. The architecture of ResNet101.

Moreover, another type of CNN-based model is called VGG, which refers to a family of deep neural network-based models. VGG models' main novelty is the use of a minuscule kernel size of 3*3 in the first hidden layers [38], which contradicts the large receptive fields in some other models such as AlexNet [39]. This small kernel is repeated throughout the network with a stride of 1 in each convolutional layer. The main advantage of this is the usage of multiple activation functions since the number of layers is bigger, which helps the model's non-linearity and makes it easier to learn more discriminative features. Fig. 4 shows the architecture of a specific type of VGG-based model named VGG16.

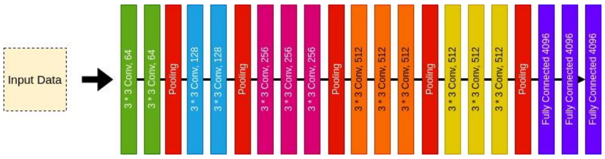


Fig. 4. The architecture of VGG16.

2.5. Transfer Learning

Transfer learning is a machine learning technique that effectively transfers knowledge developed by a network to another domain [40]. In this technique, we train a neural network on a large amount of data and then repurpose it for a different set of data. Transfer learning gives us the opportunity to ramify the problem of a small number of training samples in a specific task [41]. The most common way of knowledge transfer in the field of computer vision is to train a CNN-based model on a huge number of training images such as ImageNet and then start the training related to our specific task with parameters initialized by the pretrained values. The most notable advantage of transfer learning is that, by adopting this technique, we can achieve better results in a less number of epochs with better convergence. In this study, we adopted transfer learning with the aim of optimizing our training procedure. We trained ResNet101 and VGG16 on the images of the ImageNet dataset converted to grayscale ones.

3. EXPERIMENTAL RESULTS

3.1. Experimental Setup

In this study, we have used Python 3.7 as the programming language for implementing the proposed approach. In addition to this, the deep learning framework which is used for training and the implementation of the models is Pytorch 1.10. The machine used in this paper had Intel® Core™ i7-10450 CPU @ 2.80 GHz × 8 for its CPU and GeForce GTX 1050 for its Graphical Processing Unit (GPU).

3.2. Dataset

For training and evaluation of our proposed methodology, we used a dataset collected from two Information Technology (IT)-based companies in Iraq, named Jiasaz and Lucid. Our dataset contains 3000 malware samples and 3000 benign or non-malicious samples collected from Ubuntu 20.04.2. We have used an 80-20 policy for splitting our data into train and test sets. Table 1 shows the distribution of our data.

Table 1. The train, validation, and test distribution of the data used in this paper.

	Total	Train	Validation	Test
Non-malicious	3000	1920	480	600
Malicious	3000	1920	480	600

3.3. Hyperparameters Setting and Implementation Details

Table 2 details the list of the hyperparameters chosen in order to optimize the approach. As is seen, we have used an Adam optimizer to optimize the network's learnable parameters with a learning rate 0f 0.004.

Table 2. Hyperparameters.

Image size	256
Epoch	20
Batch size	64
Normalization standard deviation for image preprocessing	0.229
Normalization mean for image preprocessing	0.485
Optimizer	Adam
Learning rate for the optimizer	0.004
Loss function	Binary Cross-Entropy
Last layer	Sigmoid

In the training stage, we first trained the ResNet101 and VGG16 models on the grayscale version of the ImageNet dataset. After that, we freeze the first three blocks of both models and train them together in an end-to-end fashion to fine-tune our target data.

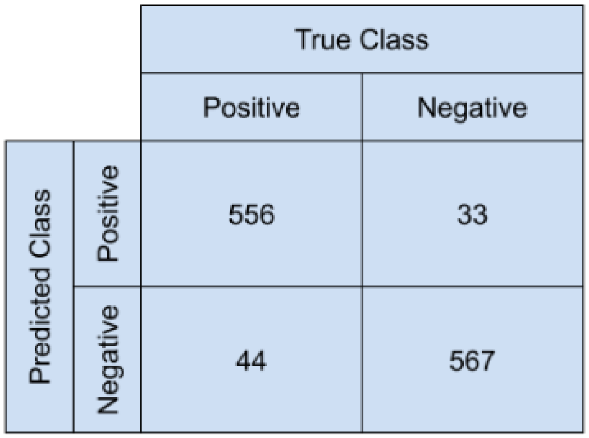
3.4. Classification Results

This section includes the results achieved by our proposed approach. Specifically, we evaluated our methodology using the metrics detailed in Table X. The metrics introduced in Table 3 are the most frequent ones used in the classification tasks and represent a trustworthy evaluation of any classifier's performance.

Table 3. Classification metrics used in this study.

Metric name	Description
True Positive (TP)	Samples that are correctly predicted as malicious
True Negative (TN)	Samples that are correctly predicted as non-malicious
False Positive (FP)	Samples that are wrongly predicted as malicious
False Negative (FN)	Samples that are wrongly predicted as non-malicious
Confusion matrix	A matrix demonstrating TP, TN, FP, and FN and used for performance analysis in classification tasks
Accuracy	$\frac{TP+TN}{TP+TN+FN+FP}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1-Score	$\frac{2* precision* recall}{precision+recall}$
AUC-ROC	The Area Under Curve for Receiver Operator Characteristic at various thresholds

Based on the metrics introduced above, Fig. 5 and Table 4 demonstrate the confusion matrix and the results achieved by our approach, respectively. Additionally, Fig. 6 and Fig. 7 show the training and validation accuracy vs. epoch curves and training and validation loss vs. epoch, respectively. Also, Fig. 8 illustrates the AUC-ROC for the proposed classifier.



		True Class	
		Positive	Negative
Predicted Class	Positive	556	33
	Negative	44	567

Fig. 5. The confusion matrix achieved by the proposed classifier.

Table 4. Classification metrics used in this study.

Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC-ROC
93.58	94.39	92.67	93.52	0.97

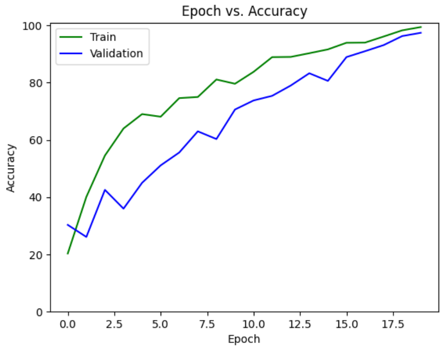


Fig. 6. Epoch vs. Accuracy for the training and validation set.

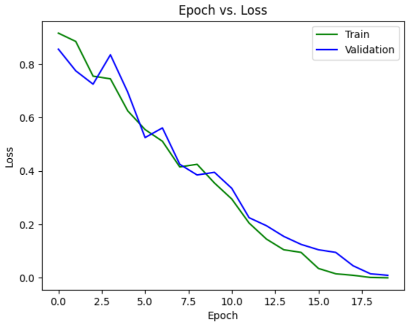


Fig. 7. Epoch vs. Loss for the training and validation set.

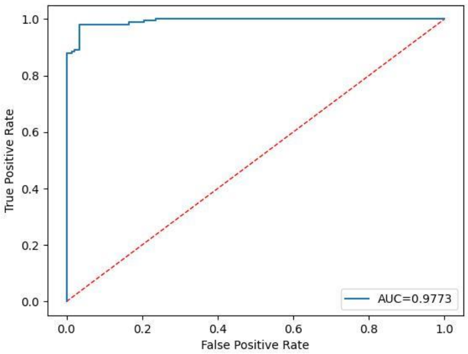


Fig. 8. AUC-ROC curve for the proposed classifier.

Moreover, to demonstrate our proposed framework's efficiency, we have compared our results with those of the previous works. As is seen in Table 5, the results achieved by our methodology are competitive in comparison with the other methodologies existing in the literature.

Table 5. Comparison with previous works,

Work	method	Accuracy	Inference time (s)
[42]	RIPPER	99.80	0.75
[43]	SVM	97.02	130
[44]	DNN	89.10	139
[45]	PSI-Graph	98.70	107
[46]	CNN	94.00	Not reported
Ours	Ensemble CNNs	93.58	0.012

The main advantage of our proposed method is its simplicity and minimalistic approach in the training phase. Our approach proves the performant outcome of transfer learning in the domain of malware image classification in that a small number of epochs suit the challenges at hand. This shows that the use of transfer learning can provide us with an optimized way of training for the challenging task of grayscale images obtained by preprocessing the binary malware files. Furthermore, the proposed model contains less trainable parameters since most layers are frozen in the main training phase.

4. CONCLUSION

This paper proposes an approach based on a combination of two CNN models to detect malware. To implement our approach, we have used ResNet101 and VGG16 as the feature extractor and employed transfer learning to have a better training process. We have conducted our evaluation on a dataset of 6000 samples equally chunked into malicious and benign classes. Our experiments demonstrate competitive and satisfactory results, proving the proposed approach's efficacy for being used in real IoT-based systems as a measurement for detecting and preventing harmful intruders and attacks.

REFERENCES

[1] Salman O, Elhajj I, Chehab A, Kayssi A, **“IoT survey: An SDN and fog computing perspective”** *Computer Networks*. Vol. 143, pp. 221-46, 2018.

[2] Talal H, Zagrouba R, **“MADS Based on DL Techniques on the Internet of Things (IoT): Survey”** *Electronics*. Vol. 10(21), 2598, 2021.

[3] Bagane P, Joseph SG, Singh A, Shrivastava A, Prabha B, Shrivastava A, **“Classification of Malware using Deep Learning Techniques. In2021 9th International Conference on Cyber and IT Service Management (CITSM)”** , pp. 1-7, *IEEE*, 2021.

[4] Maraveas C, Piromalis D, Arvanitis KG, Bartzanas T, Loukatos D, **“Applications of IoT for optimized greenhouse environment and resources management”** *Computers and Electronics in Agriculture*. Vol. 198, 106993, 2022.

[5] Mohindru V, Vashishth S, Bathija D, **“Internet of Things (IoT) for Healthcare Systems: A Comprehensive Survey”** *Recent Innovations in Computing*. pp. 213-29, 2022.

[6] Phasinam K, Kassanuk T, Shinde PP, Thakar CM, Sharma DK, Mohiddin M, Rahmani AW, **“Application of IoT and cloud computing in automation of agriculture irrigation”** *Journal of Food Quality*. 2022.

[7] Wu Y, Dai HN, Wang H, Xiong Z, Guo S, **“A survey of intelligent network slicing management for industrial IoT: integrated approaches for smart transportation, smart energy, and smart factory”** *IEEE Communications Surveys & Tutorials*. 2022.

[8] Li C, Cao Z, **“Lora networking techniques for large-scale and long-term IoT: A down-to-top survey”** ACM *Computing Surveys (CSUR).* Vol. 55(3), pp. 1-36, 2022.

[9] Al-Hadhrami Y, Hussain FK, **“DDoS attacks in IoT networks: a comprehensive systematic literature review”** *World Wide Web*. Vol. 24(3), pp. 971-1001, 2021.

[10] Liang X, Kim Y, **“A survey on security attacks and solutions in the IoT network”** *In2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC),* pp. 0853-0859, IEEE, 2021.

[11] Mishra N, Pandya S, **“Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review”** *IEEE Access*. Vol. 9, pp. 59353-77, 2021.

[12] Bahaa A, Abdelaziz A, Sayed A, Elfangary L, Fahmy H, **“Monitoring real-time security attacks**

for IoT systems using DevSecOps: a systematic literature review" *Information*. Vol. 12(4), 154, 2021.

[13] Raju AD, Abualhaol IY, Giagone RS, Zhou Y, Huang S, **"A survey on cross-architectural IoT malware threat hunting"** *IEEE Access*. Vol. 9, 91686-709, 2021.

[14] Dahl GE, Stokes JW, Deng L, Yu D, **"Large-scale malware classification using random projections and neural networks"** *In2013 IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 3422-3426, 2013.

[15] Firdausi I, Erwin A, Nugroho AS, **"Analysis of machine learning techniques used in behavior-based malware detection"** *In2010, second international conference on advances in computing, control, and telecommunication technologies,* pp. 201-203, 2010.

[16] Nguyen HT, Ngo QD, Le VH, **"A novel graph-based approach for IoT botnet detection"** *International Journal of Information Security*. Vol. 19(5), pp. 567-77, 2020.

[17] Alasmary H, Khormali A, Anwar A, Park J, Choi J, Abusnaina A, Awad A, Nyang D, Mohaisen A, "**Analyzing and detecting emerging Internet of Things malware: A graph-based approach"** IEEE *Internet of Things Journal*. Vol. 6(5), pp. 8977-88, 2019.

[18] Majid AA, Alshaibi AJ, Kostyuchenko E, Shelupanov A, **"A review of artificial intelligence-based malware detection using deep learning"** *Materials Today: Proceedings*. 2021.

[19] Fukushima, K, **"Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unafected by shift in position"** *Biol. Cybern*. 36, pp. 193–202, 1980.

[20] Ouhami M, Hafiane A, Es-Saady Y, El Hajji M, Canals R, **"Computer vision, IoT and data fusion for crop disease detection using machine learning: A survey and ongoing research"** *Remote Sensing*. Vol. 13(13), pp. 2486, 2021.

[21] Ouhami M, Hafiane A, Es-Saady Y, El Hajji M, Canals R, **"Computer vision, IoT and data fusion for crop disease detection using machine learning: A survey and ongoing research"** *Remote Sensing*. Vol. 13(13), 2486, 2021.

[22] Zaidi SS, Ansari MS, Aslam A, Kanwal N, Asghar M, Lee B, **"A survey of modern deep learning based object detection models"** *Digital Signal Processing*. 103514, 2022.

[23] Yang G, Ye Z, Zhang R, Huang K, **"A comprehensive survey of zero-shot image classification: methods, implementation, and fair evaluation"** *Applied Computing and Intelligence*. Vol. 2(1), pp. 1-31, 2022.

[24] Rasti B, Chang Y, Dalsasso E, Denis L, Ghamisi P, **"Image restoration for remote sensing: Overview and toolbox"** *arXiv preprint arXiv*:2107.00557. 2021.

[25] Jaiswal T, **"Image captioning through cognitive IOT and machine-learning approaches"** *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*. Vol. 12(9), pp. 333-51, 2021.

[26] Nourani V, Molajou A, Najafi H, Danandeh Mehr A, **"Emotional ANN (EANN): a new generation of neural networks for hydrological modeling in IoT"** *InArtificial intelligence in IoT,* pp. 45-61, Springer, Cham, 2019.

[27] Xu R, Lv P, Xu F, Shi Y, **"A survey of approaches for implementing optical neural networks"** *Optics & Laser Technology*. Vol. 136, 106787, 2021.

[28] Gunavathi C, Sivasubramanian K, Keerthika P, Paramasivam C, **"A review on convolutional neural network based deep learning methods in gene expression data for disease diagnosis"** *Materials Today: Proceedings*. Vol. 45, pp. 2282-5, 2021.

[29] Sarker IH, **"Deep cybersecurity: a comprehensive overview from neural network and deep learning perspective"** *SN Computer Science*. Vol. 2(3), pp. 1-6, 2021.

[30] He K, Zhang X, Ren S, Sun J, **"Deep residual learning for image recognition"** *InProceedings of the IEEE conference on computer vision and pattern recognition,* pp. 770-778, 2016.

[31] Simonyan K, Zisserman A, **"Very deep convolutional networks for large-scale image recognition"** *arXiv preprint arXiv*:1409.1556. 2014.

[32] Li B, Lima D, **"Facial expression recognition via ResNet-50"** *International Journal of Cognitive Computing in Engineering*. Vol. 2, pp. 57-64, 2021.

[33] Lu L, Jin P, Pang G, Zhang Z, Karniadakis GE, **"Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators"** *Nature Machine Intelligence*. Vol. 3(3), pp. 218-29, 2021.

[34] Roelofs R, Shankar V, Recht B, Fridovich-Keil S, Hardt M, Miller J, Schmidt L, **"A meta-analysis of overfitting in machine learning"** *Advances in Neural Information Processing Systems*. 2019;32.

[35] Chen CT, Gu GX, **"Generative deep neural networks for inverse materials design using backpropagation and active learning"** *Advanced Science*. Vol. 7(5), 1902607, 2020.

[36] Liu T, Chen M, Zhou M, Du SS, Zhou E, Zhao T, **"Towards understanding the importance of shortcut connections in residual networks"** *Advances in neural information processing systems*. 2019;32.

[37] Seo S, Rim DJ, Lim M, Lee D, Park H, Oh J, Kim C, Kim JH, **"Shortcut connections based deep speaker embeddings for end-to-end speaker verification system"** *system*. Vol. 13(15):17, 2019.

[38] Sitaula C, Hossain MB, **"Attention-based VGG-16 model for COVID-19 chest X-ray image classification"** *Applied Intelligence*. Vol. 51(5), pp. 2850-63, 2021.

[39] Krizhevsky A, Sutskever I, Hinton GE, **"Imagenet classification with deep convolutional neural networks"** *Advances in neural information processing systems*. 2012;25.

[40] Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, Xiong H, He Q, **"A comprehensive survey on transfer learning"** *Proceedings of the IEEE*. Vol.

109(1), pp. 43-76, 2020.

[41] Neyshabur B, Sedghi H, Zhang C, **“What is being transferred in transfer learning?”** *Advances in neural information processing systems*. Vol. 33, pp. 512-23, 2020.

[42] F. Shahzad, M. Farooq, Elf-miner, **“Using structural knowledge and data mining methods to detect new (linux malicious executables”** *Knowledge and Information Systems*, Vol. 30, No. 3, pp. 589-612, 2012.

[43] Phu, T.N., Hoang, L.H., Toan, N.N., Tho, N.D. and Binh, N.N., CFDVex, **“A Novel Feature Extraction Method for Detecting Cross-Architecture IoT Malware”** *In Proceedings of the Tenth International Symposium on Information and Communication Technology*, pp. 248-254, 2019.

[44] Jiawei Su, Danilo Vasconcellos Vargas, Sanjiva Prasad, Daniele Sgandurra, Yaokai Feng, Kouichi Sakurai, **“Lightweight Classification of IoT Malware based on Image Recognition**” 2018 *IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC*), Vol. 2, pp. 664-669, 2018.

[45] Ngo QD, Nguyen HT, Le VH, Nguyen DH, **“A survey of IoT malware and detection methods based on static features”** *ICT Express*. Vol. 6(4), pp. 280-6, 2020.

[46] Su J, Vasconcellos DV, Prasad S, Sgandurra D, Feng Y, Sakurai K, **“Lightweight classification of IoT malware based on image recognition”** *In2018 IEEE 42Nd annual computer software and applications conference (COMPSAC)*, Vol. 2, pp. 664-669, IEEE, 2018.