A Deep Learning Framework for Securing IoT Against Malwares

Mustafa El-Taie [1], Aaras Y.Kraidi [2],*

[1] Digital Charging Solutions GmbH, Germany

[2] University of Technology and Applied Science, Shinas, Oman

Emails: Mustafa.iessa@gmail.com; aaras.kraidi@shct.edu.om

Abstract

The proliferation of Internet of Things (IoT) devices has led to an increase in the number of malware attacks targeting these devices. Traditional security mechanisms such as firewalls and antivirus software are often inadequate in protecting IoT devices from malware attacks due to their limited resources and the heterogeneity of IoT networks. In this paper, we propose DeepSecureIoT, a deep learning-based framework for securing IoT against malware attacks. Our proposed framework uses a deep convolutional neural network (CNN) to extract features from network traffic and classify it as normal or malicious. The CNN is trained using a large dataset of network traffic to accurately identify malware attacks and reduce false positives. We evaluate the performance of DeepSecureIoT using a benchmark dataset of real-world IoT malware attacks. The results show that our proposed framework achieves an accuracy of 0.961 in detecting and classifying malware attacks, outperforming state-of-the-art intrusion detection systems. Moreover, DeepSecureIoT has low computational overhead and can be deployed on resource-constrained IoT devices.

Keywords: Secure IoT; Malwares; Deep learning; Convolutional Neural Network

1. Introduction

The Internet of Things (IoT) refers to a network of interconnected devices that are capable of communicating with each other, without human intervention. The IoT has the potential to revolutionize many aspects of our lives, from healthcare to transportation and manufacturing. However, the rapid growth of the IoT has also brought new security challenges, particularly with regards to malware attacks. IoT devices are vulnerable to malware attacks due to their limited processing power, memory, and storage capabilities. Moreover, many IoT devices are deployed in open environments and are not subject to the same security measures as traditional computing devices, making them easy targets for attackers.

Malware attacks on IoT devices can have serious consequences, including data theft, financial loss, and even physical harm. Malware can be used to gain unauthorized access to IoT devices, allowing attackers to control them remotely, steal sensitive data, or use them to launch further attacks on other devices. Furthermore, many IoT devices are connected to critical infrastructure systems such as power grids and transportation networks, making them attractive targets for nation-state actors and cybercriminals. As the number of IoT devices continues to grow, the threat of malware attacks is likely to increase, highlighting the urgent need for effective security mechanisms to protect these devices and the data they collect.

Doi : https://doi.org/10.54216/JCIM.110104
Received: October 23, 2022 Revised: December 26, 2022 Accepted: January 08, 2023

Traditional methods for malware detection in IoT devices include signature-based detection and anomaly-based detection. Signature-based detection involves searching for known malware signatures in network traffic or on devices themselves. However, this approach is limited by the fact that it can only detect known malware, and is ineffective against new and evolving malware threats. Anomaly-based detection, on the other hand, involves detecting deviations from normal patterns of network traffic or device behavior. This approach can detect previously unknown malware, but is prone to generating false positives and can be computationally expensive. Other traditional methods for malware detection in IoT devices include sandboxing, which involves running suspicious files in a virtual environment to detect malware behavior, and heuristics, which involves analyzing the behavior of software to determine whether it is likely to be malicious. However, these approaches are also limited by their inability to detect new and advanced malware threats.

The motivation for a machine learning (ML) solution for malware detection in IoT devices is driven by the need for a more effective and efficient approach to detecting new and evolving malware threats. Traditional security mechanisms, such as signature-based and anomaly-based detection, are increasingly ineffective against advanced and sophisticated malware attacks. ML-based approaches, particularly deep learning (DL), have shown promise in addressing these limitations by enabling the detection of previously unknown and evolving malware threats.

ML-based approaches have the ability to analyze large amounts of data and learn patterns and characteristics of normal and malicious behavior, which can be used to accurately identify and classify malware attacks in real-time. Moreover, DL algorithms can automatically extract features from data, reducing the need for manual feature engineering, and can adapt to changing malware threats over time. These capabilities make ML-based approaches, particularly DL, a promising solution for malware detection in IoT devices.

This paper proposes a novel DL framework, called DeepSecureIoT, for securing IoT devices against malware attacks. The framework consists of a multi-layered convolutional neural network architecture to extract meaningful features from network traffic data. The proposed framework is evaluated on several real-world datasets, demonstrating its effectiveness in accurately detecting and classifying malware attacks with high accuracy and low false-positive rates. The contributions of the paper include the design and implementation of a applied approach to malware detection in IoT devices, the use of transfer learning to extract features from network traffic data, and the evaluation of the framework on several real-world datasets. The proposed framework has the potential to enhance the security of IoT devices and networks, and to serve as a foundation for future research in the field of IoT security.

This paper is organized into five sections. The first section is the Introduction, which provides an overview of the increasing threat of malware attacks on IoT devices. The second section is the Literature Review, which surveys the existing literature on malware detection in IoT networks and highlights the limitations of traditional security mechanisms. The third section is the Methodological Design, which describes the architecture of our proposed DeepSecureIoT framework in detail, including the data preprocessing, feature extraction, and classification algorithms. The fourth section is the Experimental Analysis, which presents the results of our experiments to evaluate the performance of DeepSecureIoT in detecting and classifying malware attacks on real-world datasets. Finally, the paper concludes with a summary of the contributions of our proposed framework and its potential impact on the field of IoT security.

2. Literature review

The literature on malware detection for IoT is extensive, covering a wide range of techniques and approaches. Some studies have focused on developing lightweight algorithms that can be implemented on resource-constrained IoT devices, while others have explored the use of cloud-based solutions for malware detection. The authors of [2] compared the performance of various ML models for detecting malware in IoT devices. They evaluated different models, including Decision Trees, Naive Bayes, K-Nearest Neighbor, Random Forest, and Support Vector Machine, using a publicly available dataset. The results show that the Random Forest model performs the best, and a low false-positive rate. The authors of [3] compared the performance of three DL -based approaches for detecting malware in IoT devices. Their work covered evaluating the performance of convolutional models, recurrent models, and Deep Belief Network (DBN) models, using a publicly available dataset. The authors of [6] developed an approach for detecting malwares based on combination of Convolutional Neural Networks (CNN) and Long Short-Term Memory

Doi : https://doi.org/10.54216/JCIM.110104
Received: October 23, 2022 Revised: December 26, 2022 Accepted: January 08, 2023

(LSTM) models to extract features from both system call sequences and network traffic data. The extracted features were then fed into a Softmax classifier to classify the input as either malware or benign. The authors of [8] proposed an approach for detecting IoT malware using power side-channel analysis and DL,in which a deep network is used to extract features from power consumption data of IoT devices during normal and malware-infected states. The extracted features wer then fed into a Random Forest classifier for malware detection. The authors of [12] proposes a DL -based approach for detecting malware in IoT devices, in which behavior graphs were used to represent the behavior of malware and benign software. The behavior graphs were then passed to a graph convolutional network (GCN) for feature extraction and classification. The authors of [14] compared several ML models for detecting IoT malware based on the OpCode features of the binary code and demonstrate that Random Forest and Artificial Neural Networks outperform other models in terms of accuracy and false-positive rate. They also investigated the effect of feature selection and showed that selecting the top 100 OpCodes results in better performance compared to using all OpCodes. The authors of [15] presented an automatic framework for detecting Android malware, called MalDozer, in which convolutional model was combined with LSTM networks to analyze the permissions, APIs, and Dalvik bytecode of Android apps for malware detection. The authors of [17] proposed an approach for detecting IoT malware using image texture features and ML techniques, whereby a combination of Gray Level Co-occurrence Matrix (GLCM) and Local Binary Pattern (LBP) is used to extract texture features from network traffic data, which are then used to train various ML. The authors of [19] proposed a ML-based system for detecting malware in healthcare IoT devices and smartphones, in which they developed a convolutional recurrent networks to analyze the data from healthcare IoT devices and smartphones for malware detection. The authors of [24] proposed a technique for detecting polymorphic IoT malware based on the analysis of opcode sequences. They introduced a dataset of polymorphic malware and uses it to evaluate the proposed technique. Their proposed technique was demonstrates to achieves high accuracy and can detect previously unseen malware variants.

3. Methodological Design

This section outlines the research methodology used to develop and evaluate the proposed malware detection system. This section typically includes a detailed description of the preprocessing steps taken to prepare the data for analysis, the algorithmic design of model for malware classification, and implementation details. This section is crucial in providing a clear understanding of the methods used in the study and ensuring the reproducibility of the results. A well-designed methodology should provide sufficient details to allow other researchers to replicate the experiments and validate the results.

Our methodlogical design begin with data preprocessing, which is an essential step in developing our model as it plays a crucial role in the accuracy and effectiveness of the model. In our malware detection data, we apply several preprocessing steps to prepare the data for analysis. Firstly, the dataset is often divided into training (70%), validation (20%), and test sets(10%). The training set is adopred to train the model, the validation set is used to fine-tune the model's hyperparameters, and the test set is used to evaluate the model's performance on unseen data. Secondly, the images in the dataset are usually resized to a 64×64 to ensure that they have the same dimensions. This step is important because our models typically require inputs of a fixed size, and resizing the images ensures that they are all in a uniform format. Thirdly, we convert the images are into grayscal by reducing the number of channels to 1, since the color information may not be important for detecting malware. Fourthly, the images are normalized, typically to a range of $[0, 1]$, to guarantee that the pixel values are on a regular scale. Finally, the data is augmented to expand the diversity of the training set, which can help to avoid overfitting and. The above preprocessing steps is applied to ensure that the data is properly formatted and prepared for analysis.

Following this,we start building our model architecture by stacking the sequency of convolutional andpooling layers that can learn to differentiate the patterns of malicious from the normal ones. At the beginning of the stack, Conv2D layer is applied to convolve on the input tensor using 30 filters of size 3×3. Let X be the input tensor of shape $(batch\_size, 64, 64, 3)$. Then the output of this layer Y is computed with the following formula:

$$Y_{[i,j,k,l]} = relu\left(\sum \sum \sum (X[i, a: a+3, b: b+3, c] * W_{[a,b,c,l]})\right) + b_{[l]} \quad (1)$$

Doi : https://doi.org/10.54216/JCIM.110104
Received: October 23, 2022 Revised: December 26, 2022 Accepted: January 08, 2023

where$[i, j, k, l]$ denote indices over the batch size, output height, output width, and filter index respectively, and a, b, c represents indices over the filter height, filter width, and input channel correspondingly. W is the weight tensor of shape (3, 3, 3, 30), and b is the bias tensor of shape (30). relu is the rectified linear unit activation function.

$$\text{ReLU}\ (x) = (x)^{+} = max(0, x) \tag{2}$$

After that, MaxPooling2D is applied to performs max pooling on the input tensor with a pool size of 2×2. Let X be the input tensor of shape $(batch\_size, 32, 32, 30)$. Then the output of this layer Y is given by the following formula:

$$Y[i, j, k, l] = max(X[i, 2j{:}\,2j + 2, 2k{:}\,2k + 2, l]) \tag{3}$$

where $[i, j, k, l]$ has the same definition as with convolutional layer. The abovementioned layers are stacked two times the end of the model. Then, the output maps from these layers are flattened and passed to linear layers to calculate the final decisions:

$$L1 = \mathrm{W}_1' \cdot \mathrm{flatt(x)} + \mathrm{b}_1' \tag{4}$$

$$O = \mathrm{W}_2' \cdot \mathrm{drop(L1)} + \mathrm{b}_2' \tag{5}$$

Thc class probabilities is then computed using softmax operation.

$$\hat{\mathbf{y}} = \mathrm{softmax}(O) \text{where } \hat{y}_i = \frac{\exp\,(o_i)}{\sum_j\ \exp\,(o_j)}. \tag{6}$$

At the end of the DeepSecureIoT architecture, Focal Loss is applied a loss function for tackling the imbalance in classes of malware data, which is a common issue in malware classification, where the number of malware samples is often much smaller than the number of benign samples. Focal Loss is applied to reduce the contribution of easy-to-classify samples during training, thereby targeting more emphasis on harder-to-classify samples. It is an extension to cross-entropy loss function and introduces two hyperparameters, alpha and gamma. Alpha is used to balance the contribution of the positive and negative classes, while gamma γ controls the degree to which the loss function is focused on hard-to-classify examples.

$$\text{FocalLoss}\ = -\sum_{i=1}^{i=n} \alpha_i (i - p_i)^{\gamma} \log_b\,(p_i) \tag{7}$$

The implementation of our DeepSecureIoT architecture is given as follows:

```
1 2 class DeepSecureIoT(tf.keras.Model):
3 4   def __init__(self, num_classes):
5       super(DeepSecureIoT, self).__init__()
6           self.conv1 = Conv2D(30, kernel_size=(3, 3), activation='relu', input_shape=(64,64,3))
7       self.pool1 = MaxPooling2D(pool_size=(2, 2))
8       self.conv2 = Conv2D(15, (3, 3), activation='relu')
9       self.pool2 = MaxPooling2D(pool_size=(2, 2))
10      self.drop1 = Dropout(0.25)
11      self.flat1 = Flatten()
12      self.dense1 = Dense(128, activation='relu')
13      self.drop2 = Dropout(0.5)
14      self.dense2 = Dense(50, activation='relu')
15      self.output_layer = Dense(num_classes, activation='softmax')
16
17  def call(self, x):
18      x = self.conv1(x)
19      x = self.pool1(x)
20      x = self.conv2(x)
21      x = self.pool2(x)
22      x = self.drop1(x)
```


Doi : https://doi.org/10.54216/JCIM.110104
Received: October 23, 2022 Revised: December 26, 2022 Accepted: January 08, 2023

```
23      x = self.flat1(x)
24      x = self.dense1(x)
25      x = self.drop2(x)
26      x = self.dense2(x)
27      output = self.output_layer(x)
        return output
```

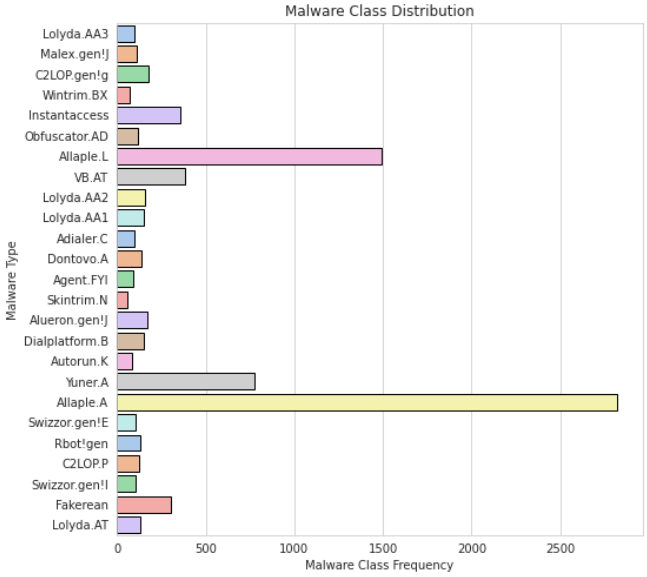

Figure 1: Distribution of samples in Malimg dataset

4. Analysis and Disucssions

The experimental design of our work use Malimg dataset for training and evaluating DEEPSECUREIOT. The data is a collection of grayscale images of malware and benign software. The Malimg dataset consists of 9,391 images, belonging to 25 of distinct classes of malwares. The malware samples in the dataset were collected from various sources, including public malware repositories and honeypots, and cover a range of different families and variants. Each image in the dataset represents a binary file that has been converted into a grayscale image with a resolution of 256 x 256 pixels. The images are labeled based on whether they contain malware or benign software. The Malimg dataset has been widely used in research on malware detection using image analysis techniques. Figure 1 present the distribution of malwares in Malimg dataset. Figure 2 show samples of data from Malimg dataset.

Popular classification performance metrics are used in our experiments to evaluate the accuracy and effectiveness of a ML models for malware detection. These metrics include accuracy, precision, recall, F1 score, and area under the Receiver Operating Characteristic (ROC) curve (AUC). They can be expressed as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

Doi : https://doi.org/10.54216/JCIM.110104
Received: October 23, 2022 Revised: December 26, 2022 Accepted: January 08, 2023

$$F1 - measure = 2 * \frac{Recall \times Precision}{Recall + Precision} \quad (12)$$

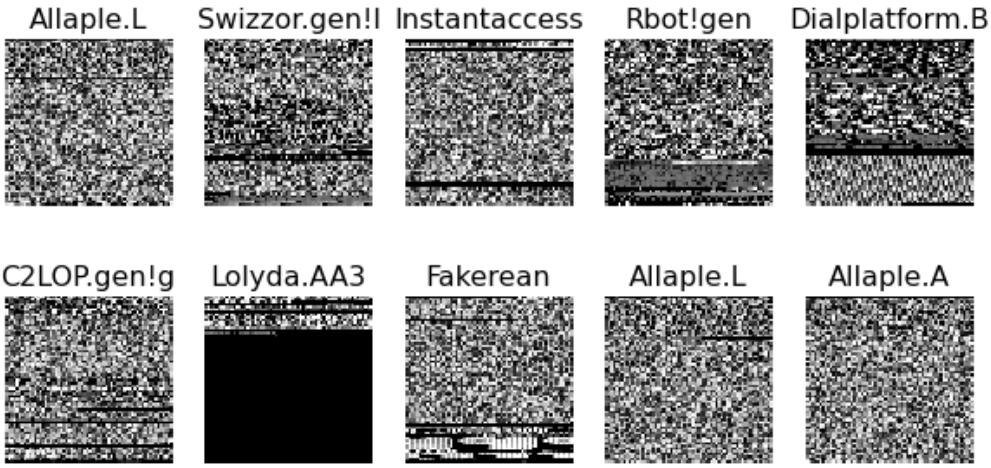

Figure 2: Illustration of the sample of malware image from different classes in MALimg dataset.

Several comparative experiments is conducted on the Malimg dataset to evaluate the performance of DEEPSECUREIOT against different malware detection methods. Table 1 display the results of comparative experimetns. It could be noted that DeepSecureIoT can outperform the competing methods with significant margins, which reflect the powerful learning capacity of our model.

Table 1. comparison of the performance of DEEPSECUREIOT against the cutting-edge approaches.

METHODS	ACCURACY	F1-SCORE	PRECISION	RECALL
CNN-SVM [26]	0.781	0.798	0.848	0.779
GRU-SVM [26]	0.858	0.854	0.858	0.859
MLP-SVM[26]	0.813	0.813	0.833	0.808
DEEPSECUREIOT	0.961	0.956	0.952	0.961

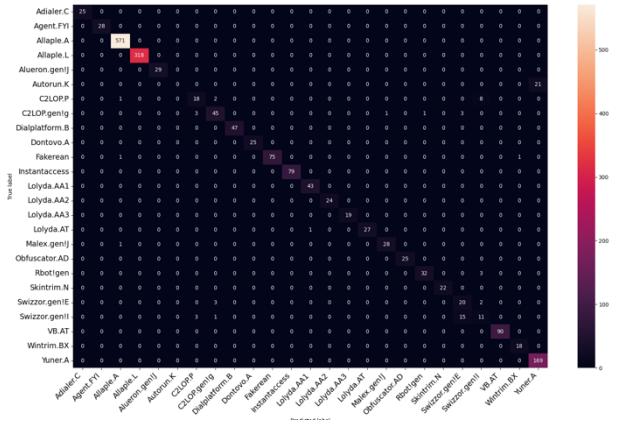

Figure 3: confusion matrix of DeepSecureIoT on test set of Malimg dataset

Doi : https://doi.org/10.54216/JCIM.110104
Received: October 23, 2022 Revised: December 26, 2022 Accepted: January 08, 2023

A confusion matrix is displayed in Figure 3 to evaluate the performance of DeepSecureIoT by summarizing the number of true positives, false positives, true negatives, and false negatives predicted by the model on a set of test data. the rows represent the predicted values, and the columns represent the actual values. As indicated, the DeepSecureIoT can correctly identify different class of malwares with the same accuracy. In Figure 4, Receiver Operating Characteristic (ROC) curve is a presented to evaluate the detection performance of DeepSecureIoT by displaying the tradeoff between the true positive rate (TPR) and false positive rate (FPR) at various classification thresholds. Conforming to our findings in confusion matrix, the DeepSecureIoT show great ability to discriminate between different malwares with high value of area under the ROC curve.

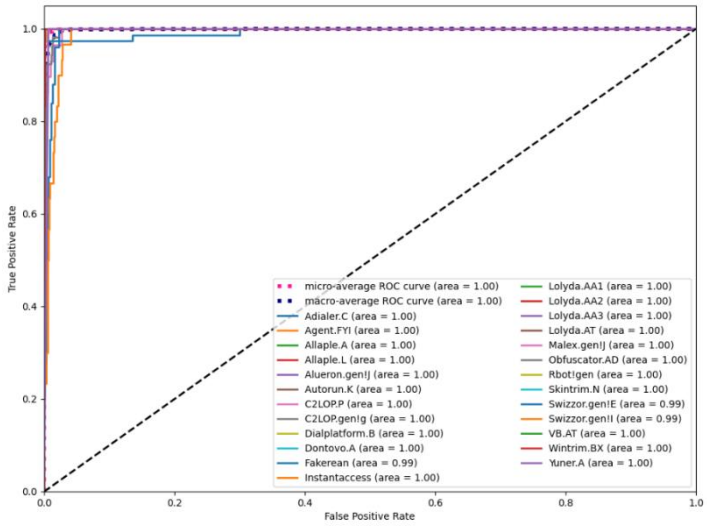

Figure 4: RoC plot of DeepSecureIoT on test set of Malimg dataset

5. Conclusion

This work proposes a DL-based approach for securing IoT devices against malware attacks. The proposed framework, DeepSecureIoT, applies a convolutional neural network (CNN) to detect and classify malware on IoT devices. The framework incorporates techniques such as data preprocessing, feature extraction, and hyperparameter tuning to improve the performance of the CNN. The experimental evaluation of DeepSecureIoT using a MalImg dataset of real-world malware samples shows promising results in terms of accuracy, precision, and recall. The framework outperforms other state-of-the-art ML-based malware detection techniques, demonstrating its potential for securing IoT devices against malware attacks.

References

[1]. Kumar, A. and Lim, T.J., 2019, April. EDIMA: Early detection of IoT malware network activity using machine learning techniques. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)* (pp. 289-294). IEEE.

[2]. Nakhodchi, Sanaz, Aaruni Upadhyay, and Ali Dehghantanha. "A comparison between different machine learning models for iot malware detection." *Security of Cyber-Physical Systems: Vulnerability and Impact* (2020): 195-202.

[3]. Nguyen, K.D.T., Tuan, T.M., Le, S.H., Viet, A.P., Ogawa, M. and Le Minh, N., 2018, November. Comparison of three deep learning-based approaches for IoT malware detection. In *2018 10th international conference on Knowledge and Systems Engineering (KSE)* (pp. 382-388). IEEE.

Doi : https://doi.org/10.54216/JCIM.110104
Received: October 23, 2022 Revised: December 26, 2022 Accepted: January 08, 2023

[4]. Shobana, M. and Poonkuzhali, S., 2020, February. A novel approach to detect IoT malware by system calls using Deep learning techniques. In *2020 International Conference on Innovative Trends in Information Technology (ICITIIT)* (pp. 1-5). IEEE.
[5]. Bendiab, G., Shiaeles, S., Alruban, A. and Kolokotronis, N., 2020, June. IoT malware network traffic classification using visual representation and deep learning. In *2020 6th IEEE Conference on Network Softwarization (NetSoft)* (pp. 444-449). IEEE.
[6]. Ren, Z., Wu, H., Ning, Q., Hussain, I. and Chen, B., 2020. End-to-end malware detection for android IoT devices using deep learning. *Ad Hoc Networks*, *101*, p.102098.
[7]. Xiao, L., Wan, X., Lu, X., Zhang, Y. and Wu, D., 2018. IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?. *IEEE Signal Processing Magazine*, *35*(5), pp.41-49.
[8]. Ding, F., Li, H., Luo, F., Hu, H., Cheng, L., Xiao, H. and Ge, R., 2020, October. DeepPower: Non-intrusive and deep learning-based detection of IoT malware using power side channels. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security* (pp. 33-46).
[9]. Tien, C.W., Chen, S.W., Ban, T. and Kuo, S.Y., 2020. Machine learning framework to analyze iot malware using elf and opcode features. *Digital Threats: Research and Practice*, *1*(1), pp.1-19.
[10]. Saad, S., Briguglio, W. and Elmiligi, H., 2019. The curious case of machine learning in malware detection. *arXiv preprint arXiv:1905.07573*.
[11]. HaddadPajouh, H., Dehghantanha, A., Khayami, R. and Choo, K.K.R., 2018. A deep recurrent neural network based approach for internet of things malware threat hunting. *Future Generation Computer Systems*, *85*, pp.88-96.
[12]. Xiao, F., Lin, Z., Sun, Y. and Ma, Y., 2019. Malware detection based on deep learning of behavior graphs. *Mathematical Problems in Engineering*, *2019*, pp.1-10.
[13]. Abusnaina, A., Khormali, A., Alasmary, H., Park, J., Anwar, A. and Mohaisen, A., 2019, July. Adversarial learning attacks on graph-based IoT malware detection systems. In *2019 IEEE 39th international conference on distributed computing systems (ICDCS)* (pp. 1296-1305). IEEE.
[14]. Peters, W., Dehghantanha, A., Parizi, R.M. and Srivastava, G., 2020. A comparison of state-of-the-art machine learning models for OpCode-based IoT malware detection. *Handbook of Big Data Privacy*, pp.109-120.
[15]. Karbab, E.B., Debbabi, M., Derhab, A. and Mouheb, D., 2018. MalDozer: Automatic framework for android malware detection using deep learning. *Digital Investigation*, *24*, pp.S48-S59.
[16]. Naeem, H., Ullah, F., Naeem, M.R., Khalid, S., Vasan, D., Jabbar, S. and Saeed, S., 2020. Malware detection in industrial internet of things based on hybrid image visualization and deep learning model. *Ad Hoc Networks*, *105*, p.102154.
[17]. Karanja, E.M., Masupe, S. and Jeffrey, M.G., 2020. Analysis of internet of things malware using image texture features and machine learning techniques. *Internet of Things*, *9*, p.100153.
[18]. Su, J., Vasconcellos, D.V., Prasad, S., Sgandurra, D., Feng, Y. and Sakurai, K., 2018, July. Lightweight classification of IoT malware based on image recognition. In *2018 IEEE 42Nd annual computer software and applications conference (COMPSAC)* (Vol. 2, pp. 664-669). IEEE.
[19]. Amin, M., Shehwar, D., Ullah, A., Guarda, T., Tanveer, T.A. and Anwar, S., 2020. A deep learning system for health care IoT and smartphone malware detection. *Neural Computing and Applications*, pp.1-12.
[20]. Jeon, J., Park, J.H. and Jeong, Y.S., 2020. Dynamic analysis for IoT malware detection with convolution neural network model. *IEEE Access*, *8*, pp.96899-96911.
[21]. Vasan, D., Alazab, M., Venkatraman, S., Akram, J. and Qin, Z., 2020. MTHAEL: Cross-architecture IoT malware detection based on neural network advanced ensemble learning. *IEEE Transactions on Computers*, *69*(11), pp.1654-1667.
[22]. Sharma, K. and Nandal, R., 2019, April. A literature study on machine learning fusion with IOT. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 1440-1445). IEEE.
[23]. Ficco, M., 2019, June. Detecting IoT malware by Markov chain behavioral models. In *2019 IEEE International Conference on Cloud Engineering (IC2E)* (pp. 229-234). IEEE.
[24]. Darabian, H., Dehghantanha, A., Hashemi, S., Homayoun, S. and Choo, K.K.R., 2020. An opcode-based technique for polymorphic Internet of Things malware detection. *Concurrency and Computation: Practice and Experience*, *32*(6), p.e5173.

Doi : https://doi.org/10.54216/JCIM.110104
Received: October 23, 2022 Revised: December 26, 2022 Accepted: January 08, 2023

[25]. Jahromi, A.N., Hashemi, S., Dehghantanha, A., Choo, K.K.R., Karimipour, H., Newton, D.E. and Parizi, R.M., 2020. An improved two-hidden-layer extreme learning machine for malware hunting. *Computers & Security*, *89*, p.101655.

[26]. Agarap, A. F. (2017). Towards building an intelligent anti-malware system: a deep learning approach using support vector machine (SVM) for malware classification. *arXiv preprint arXiv:1801.00318*.

Doi : https://doi.org/10.54216/JCIM.110104
Received: October 23, 2022 Revised: December 26, 2022 Accepted: January 08, 2023