A Method by Utilizing Deep Learning to Identify Malware Within Numerous Industrial Sensors on IoTs

Ronghua MA
Zhengzhou Railway Vocational and Technical College, Teacher Work Department of the Party Committee, Zhengzhou 450052, China

***Abstract*—The industrial sensors of IoT is an emerging model, which combines Internet and the industrial physical smart objects. These objects belong to the broad domains like the smart homes, the smart cities, the processes of the industrial and the military, the agriculture and the business. Due to the substantial advancement in Industrial Internet of Things (IIoT) technologies, numerous IIoT applications have been developed over the past ten years. Recently, there have been multiple reports of malware-based cyber-attacks targeting IIoT systems. Consequently, this research focuses on creating an effective Artificial Intelligence (AI)-powered system for detecting zero-day malware in IIoT environments. In the current article, a combined framework for the detection of the malware basis on the deep learning (DL) is proposed, that uses the dual-density discrete wavelet transform for the extraction of the feature and a combination from the convolutional neural network (CNN) and the long-term short-term memory (LSTM). The method is utilized for malware detection and classification. It has been assessed using the Malimg dataset and the Microsoft BIG 2015 dataset. The results demonstrate that our proposed model can classify malware with remarkable accuracy, surpassing similar methods. When tested on the Microsoft BIG 2015 and Malimg datasets, the accuracy achieved is 95.36% and 98.12%, respectively.**

***Keywords*—Malware; malware detection; industrial sensors; Internet of Things (IoTs); Deep Learning (DL)**

I. INTRODUCTION

The advancement of various technologies such as the sensors, the wireless communication, the embedded computing, the automatic tracking, the widespread access to the Internet and the dispensed services increases the possible of the accretion of the smart sensors in our daily lives via Internet. The convergence of Internet and the smart sensors, which can connect with together, describes IoTs. This novel example has been detected as one from the foremost significant factors on the industries of the data and the communication technology in the coming years [1].

The goal of the IoTs technology is to enable the objects for the connection at any time and any place with anything and anyone, who uses any path or any network as optimally. IoTs is a new evolution from Internet. IoTs is the new technology that pays attention to the pervasive presence of the environment and deals with the diversity of the smart objects with the wireless connections and the wired connections for the communication with together. These objects work together, to create the new applications or the new services and to achieve the common goals. In fact, they are the development challenges for the creation of a smart big world. A world that is the real, the digital and the virtual and is converging towards the formation of the smart environments. This world creates the smarter environments of the energy, the transportation, the cities health and many others [2].

However, the integration of the smart objects in the real world by Internet can bring the threats of the security in the several of our daily behaviors [3]. According to the wide standards of the communication, the limited power of the computing and the great number of the connected sensors, the common actions of the security against the threats cannot work effectively on IoTs. Therefore, the development of the specific solutions of the security for IoTs is necessary, to enable the organizations users, to detect total weaknesses of a network [4]. Several ongoing projects for evolution of the security in IoTs include the methods that provide the data confidentiality, the authentication of the control of the access on IoTs, the privacy, the trust among the users and the implementation of the security policies. [5]. Nevertheless, even with the methods, IoTs are assailable to the several attacks. The attacks which are done, to interrupt and to disrupt the networks. For this reason, the required method of the defense is the creation of the models for the detection of the attackers. The development of the web-based technologies and the cloud computing will mark the future revolution in the digital technologies. Also, it will lead to the increased health, the productivity, the convenience and a wide range of the useful information for the individuals and the organizations. On the other hand, there will be challenges in the field of the personal privacy, the complexity of the intrusion technology and the creation of a digital gap [6].

The security establishment is perhaps the biggest challenge in the IoTs network. The security in the current Internet is also considered as a big challenge, but in the Internet of Things, this issue takes on the greater dimensions. One of the reasons for this issue is the distribution of the network and the more entry points into the system. Also, the objects that are supposed to be connected to the Internet, usually have a simpler architecture than the computers, and this implementation makes the security tools as the difficult. The IoTs technology is much closer to the real life than the current Internet; In fact, the intrusion into this network will be equivalent to the intrusion into the daily life of the users [4]. Due to the security problems on the real world and in the technology of IoTs, and according to the problems of the intrusion into the networks, it is very necessary to present the optimal method, in order to discover the intrusion and to keep the security on the networks [7].

*Corresponding Author.

Following an extensive review of the literature, it has been noted that current methods face several limitations and security challenges, such as low accuracy, insufficient large datasets, limited scalability, and high prediction times for detecting zero-day malware or unknown malicious activities. Therefore, this work proposes an efficient zero-day malware detection framework utilizing a hybrid deep learning model for IIoT systems. The main contributions of this paper are as follows: i) It introduces a novel AI-powered zero-day malware detection system for IIoT, utilizing an image visualization technique by combining CNN and LSTM models. ii) D3WT is employed for deep feature extraction, breaking down malware images into approximate and detailed coefficients. iii) The proposed hybrid model is tested on three major cross-platform malware datasets and compared with advanced models. The method is applied to detect and to classify the malware. The background of research is provided in Section II. Our method is described in Section III. Section IV shows the experiments and the evaluation of the results. Section V also presents the conclusions and the effective suggestions by using the obtained findings.

II. Related Works

In the recent decade, the many models based on the theory of the game on scope of the security in the networks have been done, to model the analysis and to optimize the efficiency of IDSs in the related technology to IoTs, such as the mobile contingency networks [8, 9], WSNs [10], the cloud computing [11] and the physical cyber networks [12]. The research in [11] has presented the various intrusions, that affect the availability of the privacy and the integrity in the cloud computing. They have distributed the methods of the used IDSs in the cloud into three categories: the host-based, the network-based and the hypervisor-based. They are also reviewed the advantages and the disadvantages of every protocol and are recognized the problems, to create the cloud computing as a trustworthy architecture for the providing of IoTs. The research in [9] shows that a malware detection model is capable to handle and to control the several protocols of the communication by combining the rules of the signature and the procedures for the detection of the anomaly. The research in [10] has done a wide review on IDSs in WSNs and has provided a comparative evaluation among IDSs for WSNs, according to the architecture of the network and the method of the detection.

The research in study [13] presents a comprehensive analysis of the security from the several protocols of Internet. Specifically, the authors discuss about the security topics in IEEE 802.15.4 against 6LOWPAN, the Routing Protocol of IPv6 for RPL, the protocols of DTLS and CoAP. The research in [8] has investigated IDSs for the mobile contingency networks, by relying on the detection algorithm. A categorization basis on the tree for IDSs has been introduced according to the character of the used method for the processing in detection model. The research in [14] presents an IDS for LOWPAN-RPL6 that is capable to recognize the Sinkhole attack, the Sybile attack and the Selective attack, by using a hybrid approach that combines the different parameters. The research in study [15] has presented an IDS basis on the features with supervisor, by using forward neural network. In this paper, the feature selection is done on the ISCX-IDS 2012 dataset and the Android CIC dataset. In order to do the feature selection phase, SVM with the incremental learning has been used, which with the ranking of 43 features in the dataset, 20 features with the highest rank have been selected. Then, by using a neural network, the final detection is made with the accuracy equal to 94% and 98.7%.

The research in study [4] has presented an optimal platform, to show the possible application of the practical in the malware propagation suppression for perseverance of the privacy in the smart objects on IoTs, via an IDS by the game theory calculation of Bayesian. The research in study [16] has examined the security of IoTs, the challenges, the solutions and the threats. After checking and evaluating the possible threats and after specifying the security actions in scope of IoTs, they have done the risk analysis of the quantitative and the qualitative that examines the threats of the security on every layer. The research in [7] has investigated a new plan, by using a combination from the classical encryption and the quantum encryption, to improve the security of the Internet network. A research title which is related to the anomaly-based intrusion detection systems [3], by evaluating solutions and the researches and by using role of DL in IDS, discusses the efficiency of the proposed methods, and also, by identifying the challenge from the past researches, it recommends the deep learning-based guidelines.

The research [17], in an article, in addition to the presentation of a model based on the combination from the artificial neural networks for the intrusion detection, it provides an algorithm for extraction of the optimal features on Cup KDD, which is the standard dataset for the testing of the intrusion detection methods in the computer networks. The researches efforts in the field of IDSs for IoTs have begun and speeded up. By taking the research background, it is important to state which the presented approaches have not deeply checked the abilities and the laxities of every detection model and each placement strategy. The many authors have relied on a few kinds of the attacks. Finally, the very easy validation schemes have presented the foundation for the reproduction of the other proposed approaches.

III. The Presented Approach

Here, the details of our proposed approach are provided. This approach is described to detect the zero-day malwares along with its family by the greater accuracy for the industrial sensors of IoT. The presented method is disturbed in $three$ main steps: the first includes the data preprocessing and the image resizing, the second includes the feature extraction by using D3WT and the third includes a hybrid model of LSTM-CNN for the automatic detection of the malwares. Fig. 1 displays the framework of our presented method that is created by integrating LSTM-CNN and D3WT. The coefficients of the approximation and the detail are exploited by D3WT. The exploited features are combined as input of LSTM-CNN, to create the fused images. The proposed approach is evaluated by Microsoft BIG 2015 and Malimg, which contain the different types from the malwares. The details of this approach are provided on the below subsections.

A. Data Pre-Processing

The data preprocessing is a necessary part in every method basis on AI, in order to increase its efficiency. In the current

article, the extraction of the feature has been done from the raw dataset of the malware. Then, the obtained features (like the opcodes, the strings and the bytecodes) are converted to the digits of the binary. Next, a collection from 8 bits are converted to the grayscale image. The converted images have the various sizes in terms of the height, nevertheless, the width of the images is the constant. The conversion of this grayscale images by the bytecodes is provided by using the Python tools. In the next step, the image preprocessing (like the image resizing to a specific size equal to 224×224) is performed on this approach. Then, D3WT is used as a method for the extraction of the feature, to extract the coefficients of the approximation and the detail.

B. Extraction of Features

On our approach, D3WT is used to analyze the inputs by using the banks of the filter. It follows an iterative method. This approach includes 2 wavelets (the high pass) and a scale function (the low pass). A wavelet is the offset by another wavelet. The theoretical flow diagram for 2 filter banks of D3WT is displayed on Fig. 2. D3WT is basis on 3-channel theory for the bank of the filter of the complete reconstruction. A matrix with $three$ columns is applied for the scaling and the function of the wavelet. The filter of the scaling $\theta(\alpha)$ is placed on first column and 2 high-pass filters, which are denoted by $\varphi_1(\alpha)$ and $\varphi_2(\alpha)$, are placed on second column and the third column. The function of the scaling is denoted by $T_0(-v)$ and 2 high-pass filters are represented $T_1(-v)$ and $T_2(-v)$. The input $J(v)$ passes via the model of the filter, and the analysis operation is done by the bank of the filter for the analysis, that creates 3 sub-bands. The sub-bands are down-sampled by 2. The output of this filter is 3 signals $C(v)$, $\mathrm{D}_1(\mathrm{v})$ and $\mathrm{D}_2(\mathrm{v})$. These items are the coefficients with the low frequency (the approximation coefficient) and 2 coefficients with the high frequency (the detail coefficient), respectively [18].

The synthesis filter bank is used to inversely transform the extracted low-pass coefficient $T_0(v)$ and 2 high-pass filters $T_1(v)$ and $T_1(v)$, with the high sampling by 2, and then, in order to receive the output signal, $K(v)$ is fused. D3WT performs the iterative operation with the over-sampled filter bank, to ensure the perfect reconstruction conditions. This work leads to the transformation of the shift constants, namely $T_0(v)$, $T_0(v)$ and $T_0(v)$, which $\mathrm{K}_{\mathrm{output}}(\mathrm{v}) = \mathrm{J}_{\mathrm{Input}}(\mathrm{v})$. Also, D3WT is applied to convert the samples of the malware in the coefficients of the detail on every level from the decomposition and into the coefficients of the approximation on maximum level. 2 wavelets, namely $\varphi_1(\alpha)$ and $\varphi_1(\alpha)$, are generated to be separated by $1/2$, as shown in the following equation [19]:

$$\varphi_1(\alpha) = \varphi_2 \times (\alpha - 0.5) \quad (1)$$

The following equations are given for a multi-resolution framework, that should satisfy θ and φ_i:

$$\theta(\alpha) = \sqrt{2}\sum_v T_0(v)\theta(2\alpha - v) \quad (2)$$

$$\varphi_i(\alpha) = \sqrt{2}\sum_v T_i(v) \times \theta(2\alpha - v) \ \ where\ i = 1.2 \quad (3)$$

C. Classification of Malware Samples

Regarding CNNs, it should be said that they are used in the classification of the image for the object recognition and the classification work. The prominent amicability of CNN is according to its ability for the automatic extraction of the important features from the input samples. CNNs are a set from three important layers: convolution, pooling and fully-connected. In the layer of the convolution, the input features are checked, and the input sample is filtered. This layer does the dot product among 2 matrices, for the creation of the weight matrix and the weighted sum in the layer of the kernel. This filter does the operation of the batch among the pixel values of the input. The proper parameters for the filter size layer, the stride and the zero padding, help to increase in results of the convolution kernels. Also, ReLU is applied, to enhance the non-linearity on the map of the feature. ReLU computes the activation with the setting of the input as 0. ReLU has the values of the negative and the positive. The negatives are displayed by 0 and the positives are indicated by the max value. In the layer of the fully-connected on CNN, a classification is performed, to sequentially determine the given input of the layer of the convolutional and the pooling. In the layer of the pooling, the desired operation is applied, to reduce the dimensionality of the features, by selecting the greatest values by the area for the creation of the matrix. The ultimate layer is the fully-connected, which is applied, to flatten total features in the vectors with the single feature. It presents the communication between the neurons of the previous layer and the next layer. This point causes that feature maps from the input to the output. The output of fully-connected is taken to SoftMax, to predict the samples of the malicious [20].

Regarding LSTMs, it should be said that they include a memory unit and $three$ interactive gates: the input, the forget and the output. The memory unit is applied, to protect the late state against the prior state. The gate of the input is applied, to restrict the amount of the input data for the training in network, which is stored on the state of the unit in the time "t". The gate of the forget determines which whether the data of the input should rouse forward or take away, for the entering to the gate of the input in the time "$t - 1$". The gate of the output describes the data of the output. This gate is applied, to deal the problems of the vanishing gradient, that appear in the implementation of 3 gates [20].

In this paper, a combined model of LSTM-CNN is applied, to detect the malware. CNN is superior for the processing of the large volume from the great-dimensional datasets and for the automatic extraction of the informational features by the images. It is done by using the techniques of the optimization with the great dimensional. The efficiency of CNNs is influenced by the samples size of the training and the testing. In the dataset with the variable size, CNN requires to be fine-tuned. The goal of the use from the feature extraction is the minimization of the data dimensions and the time of the training for CNN, that leads to the improvement in accuracy of the detection. On our approach, first, D3WT is used, to extract the first level feature. The fused features include the coefficients of the approximation and the detail, which serve as the input of CNN. Then, CNN exploits the necessary informational features by the minimal dimensions. On our combined approach, 4 layers of the convolution, 4 layers of the pooling, 4 layers of the dropout, one layer of the fully-connected (the flatten, the dense and the dropout), one layer of LSTM and SoftMax are applied. The fetched features are taken as the input of LSTM (by max-pooling). Then, they are passed

to a layer of the fully-connected that transforms total features to a vector with the single feature (by SoftMax). CNN extracts the features as efficiently and as automatically. On our combined approach, CNN works as the encoder for the encoding of features (by convolution), and LSTM decodes the encoded data. Eventually, a layer of the fully-connected is applied, to classify the malware. Thus, the foremost features of 2 networks are integrated, to model an impressive combined network [21].

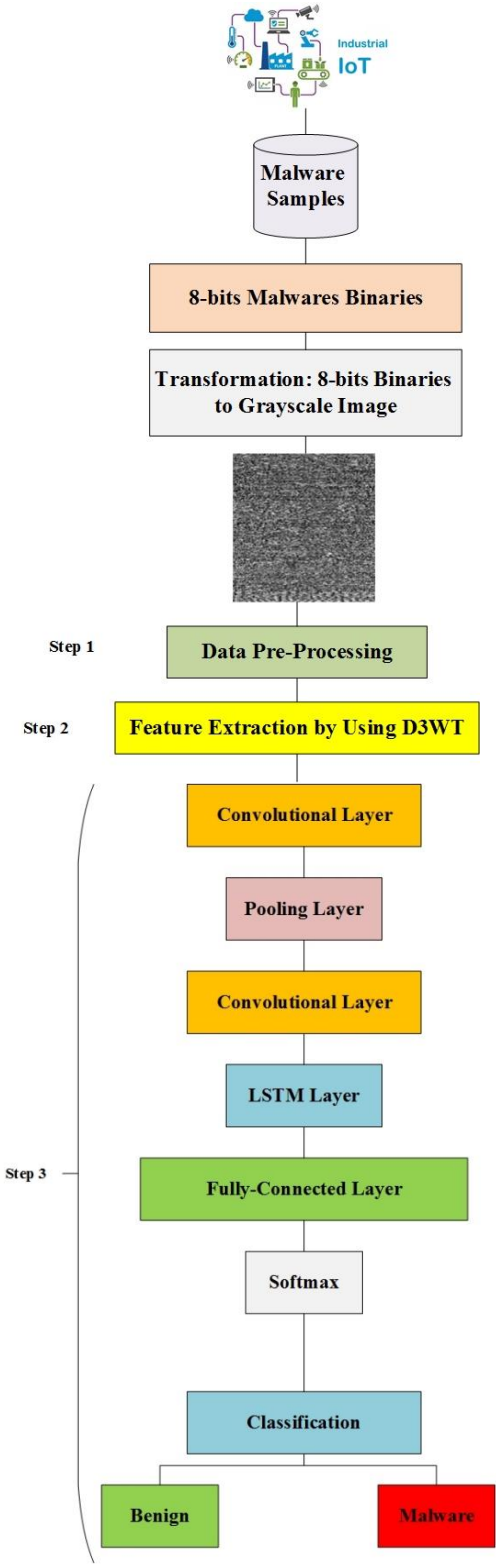

Fig. 1. The general framework of our presented method.

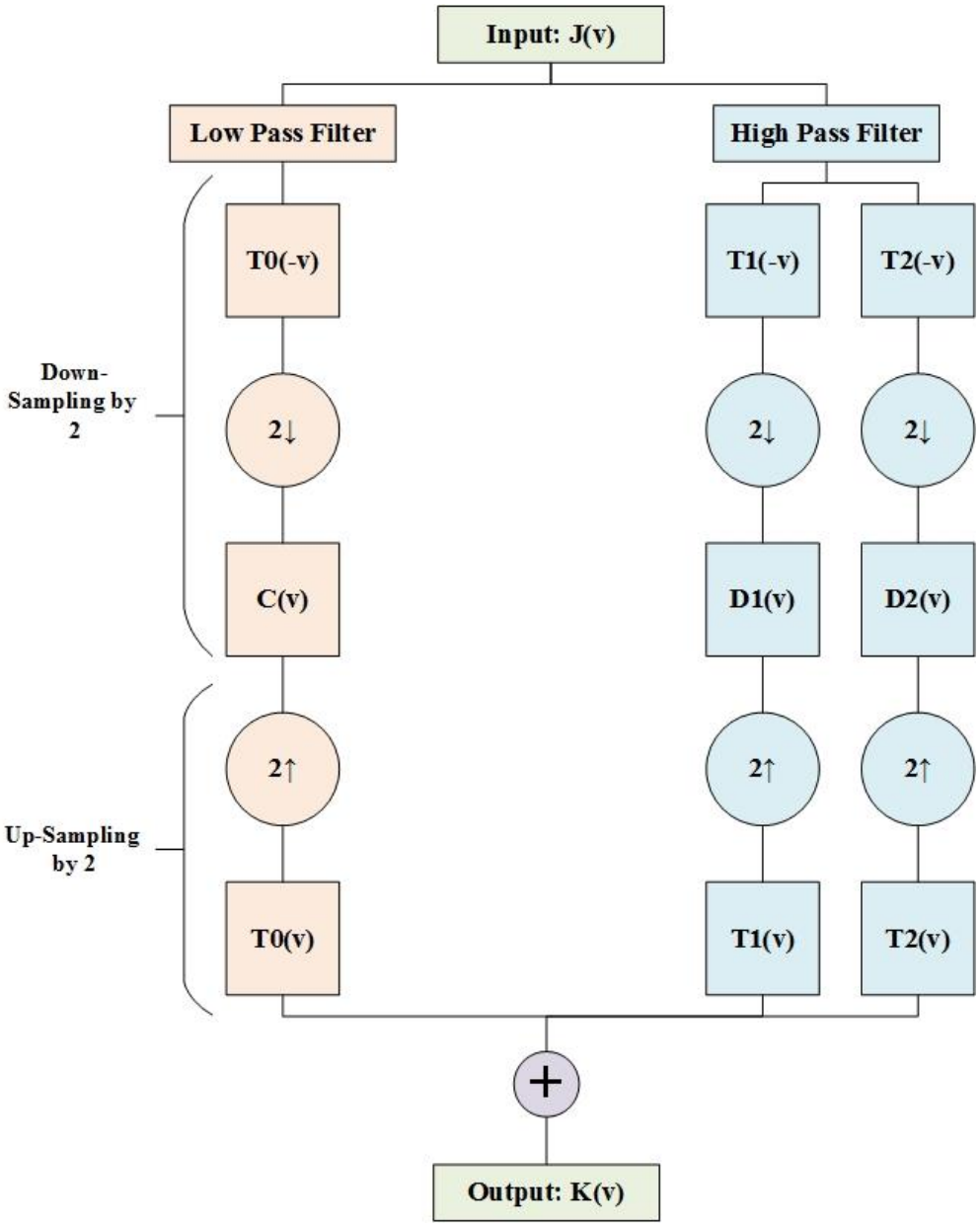

Fig. 2. The D3WT basis on 3-channel filter bank.

IV. Evaluation and Results

Here, the details of datasets and tests and results are provided. Python has been applied, to implement these tests. The proposed approach is implemented in the computer with RAM 8G and Intel(R) CPU Core(TM) i7 3.0 GHz. CNN is implemented on GPU and the graphics card is GEFORCE 840M for NVIDIA. The data of training, testing and validation are randomly selected from each dataset, and the evaluation procedures are done as one by one. In the stages of training, validation and testing, the data selection rates are set at 70%, 10%, and 20%, respectively. CNN includes 4 layers of the convolution, 4 layers of the pooling, 4 layers for the normalization of the batch and ReLU with the various sizes of the filter (32, 64 and 128). The sizes of pooling, stride and kernel are equal to 2×2, 2×2 and 3×3. After the fourth layer of the convolution, the features are passed via LSTM and the layer of the fully-connected. The layer of the fully-connected includes the layers of flatten, dense and dropout. Eventually, SoftMax applied, to classify the malware labels.

A. Datasets and Evaluation Criteria

To demonstrate the efficiency of our approach, the several criteria for the evaluation have been applied. These criteria are: sensitivity, accuracy, F1-score and specificity. These criteria are computed as the below:

$$Accuracy = \frac{TN+TP}{TN+FN+TP+FP} \quad (4)$$

$$Sensitivity = \frac{TP}{FN+TP} \quad (5)$$

$$Specificity = \frac{TN}{TN+FP} \quad (6)$$

$$F1 - Score = \frac{2\times TP}{2\times TP+FP+FN} \quad (7)$$

TP is the true positive and FP displays the false positive, while TN displays the true negative and FN displays the false negative. The mentioned criteria are the first stage for the interpretation of the efficiency for our approach. The comparison procedures are performed for the proposed combined model with 2 DNNs. These two neural networks are: AlexNet and Resnet-50.

The experiments are performed on *two* comprehensive datasets: Malimg and Microsoft BIG 2015. The dataset of Malimg [22] includes 9339 samples from the malwares. Every sample from the malwares on dataset belongs to one of 25 classes. In addition, number of the samples in a class is different. The classes of the malware are: Agent.FYI, Adialer.C, Allaple.L, Allaple.A, Alueron.gen!J, Autorun.K, Benign, C2LOP.P, C2LOP.gen!g, Dialplatform.B, Dontovo. A, Fakerean, Instantaccess, Lolyda.AA1, Lolyda.AA2, Lolyda.AA3, Lolyda.AT, Malex.gen!J, Obfuscator. AD, Rbot!gen, Skintrim.N, Swizzor.gen!E, VB.AT, Yuner.A and Wintrim.BX.

The dataset of Microsoft BIG 2015 [23] includes 21741 samples from the malwares, which are belonging to *nine* classes, and are: Lollipop, Kelihos_ver1, Ramnit, Vundo, Kelihos_ver3, Tracur, Obfuscator, Gatak, .ACY and Simda. The similar to with Malimg, the number of the samples from the malwares in the classes is not uniformly dispensed. Every sample from the malwares is displayed by 2 files: ".asm" and ".byte". In the experiments, ".byte" are only used, to form the malware images.

B. Results

In this section, the outcomes of the performance for our proposed approach and its comparison with the other models are presented. The evaluation criteria define the efficiency of the models. The acute case behind classification is a criterion for the evaluation, which is applied for the understanding of the efficiency of a model [24]. Therefore, the multiple criteria in experimental outcomes and the discussion have been used, to demonstrate the efficiency of our approach. Fig. 3 to Fig. 6 and Fig. 7 to Fig. 10 show the values of the accuracy, the sensitivity, the specificity and the F1-score for AlexNet, Resnet-50 and the presented approach on Microsoft BIG 2015 and Malimg, respectively. According to these results, it can be said which our approach works superior than the DNN architecture. Also, the performance of our approach shows the similar efficiency results on 2 datasets, while the performance of the other DNNs is significantly different on 2 datasets. The above situations show which our approach is stronger, and outperforms than 2 DNNs.

In the next step, the types of the malwares are analyzed along with the confusion matrices. Tables I, II and III show the matrices of the confusion in Microsoft BIG 2015 for *nine* types of malware by using the proposed approach and AlexNet and ResNet-50. Here, the accuracy rate for each type of the malwares is shown by using the confusion matrices. The matrix of the confusion for the proposed approach shows that it provides the better results for the entire malware classification, with the exception of vundo. In addition, the matrix of the confusion for ResNet-50, which is displayed in Table II, provides the better detection for vundo, in compared to the other approaches. In this situation, total models can quickly detect the malwares of simda and tracur.

Finally, a functional comparison with the advanced results is performed. Tables IV and V display the accuracy in Malimg and Microsoft Big 2015 by using our approach and the other models. It should be attended which the efficiency of the presented approach is better than the advanced models, insomuch it creates a greater value for the accuracy.

C. Discussion

The promising results of the proposed framework, which uses an image visualization approach, demonstrate its ability to accurately identify various malware families and new variants. These results indicate that image-based visualization is both effective and efficient for identifying malware samples across different classes. This model has significant potential for detecting advanced malware by analyzing large volumes of input data and classifying them into sub-families. However, the model's performance in training and testing may decline with smaller datasets. Additionally, inaccurate samples of both malware and benign instances can lead to poor malware identification. Despite the effectiveness of our proposed hybrid deep learning architecture in detecting and classifying various malware variants and families, there are still some limitations that need to be addressed.

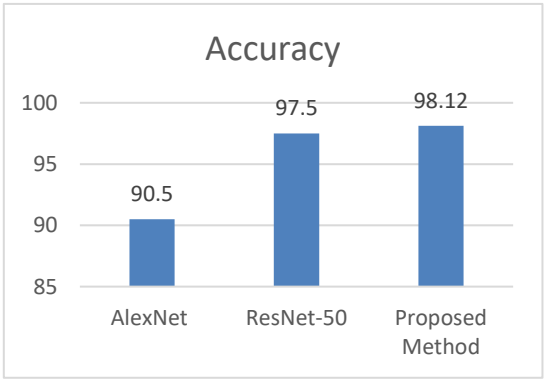

Fig. 3. The comparison of the accuracy of the various models on Malimg.

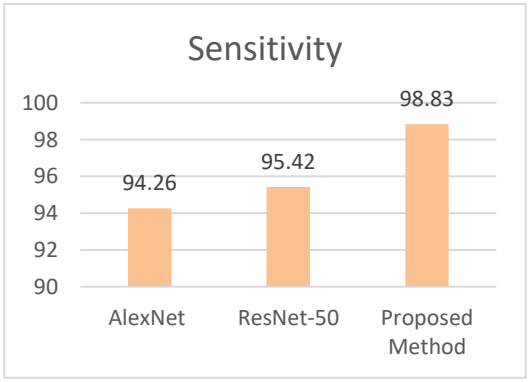

Fig. 4. The comparison of the sensitivity various models on Malimg.

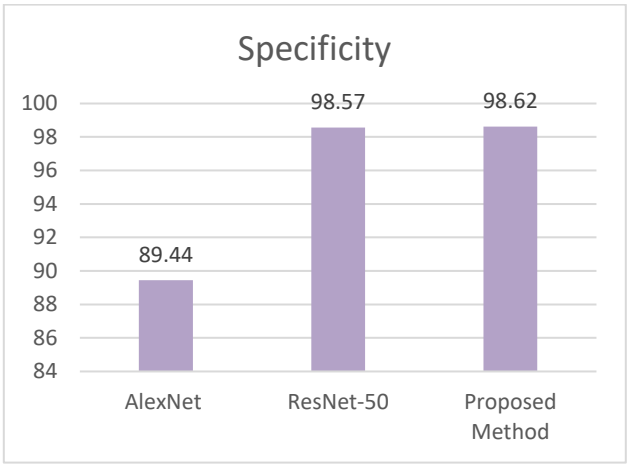

Fig. 5. The comparison of the specificity of the various models on Malimg.

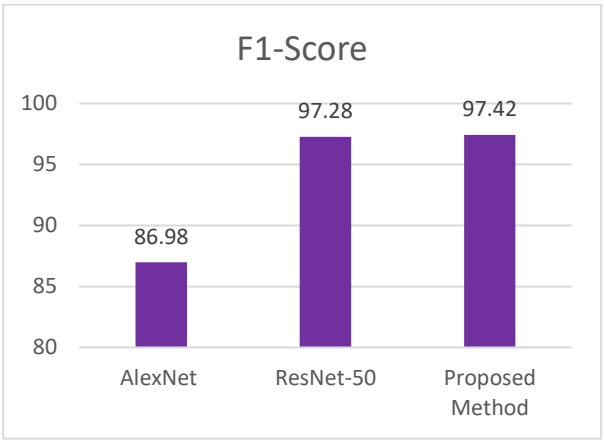

Fig. 6. The comparison of the F1-score of the various models on Malimg.

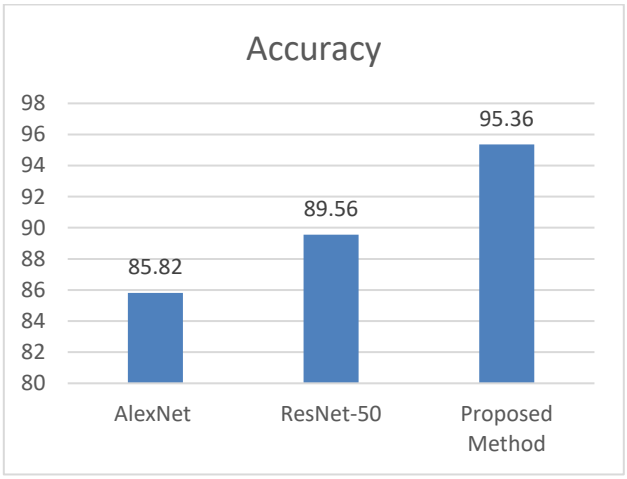

Fig. 7. The comparison of the accuracy of the various models on Microsoft BIG 2015.

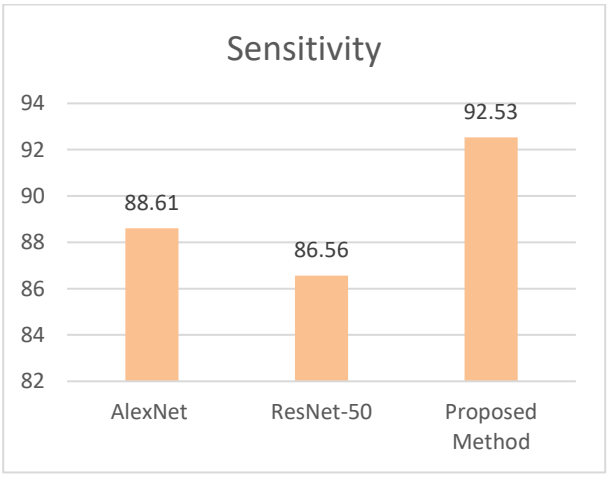

Fig. 8. The comparison of the sensitivity various models on Microsoft BIG 2015.

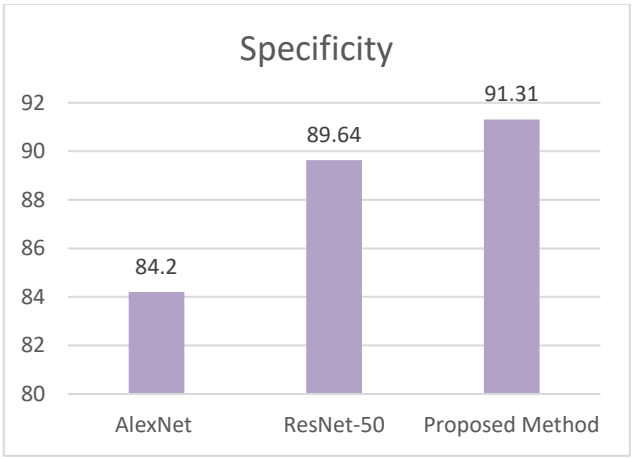

Fig. 9. The comparison of the specificity of the various models on Microsoft BIG 2015.

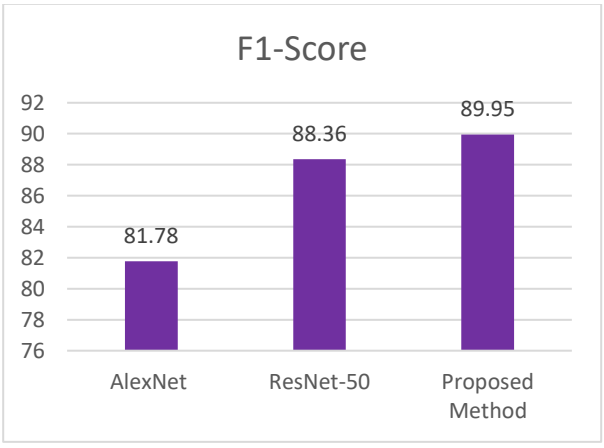

Fig. 10. The comparison of the F1-score of the various models on Microsoft BIG 2015.

TABLE I. THE MATRIX OF THE CONFUSION IN MICROSOFT BIG 2015 FOR NINE TYPES OF THE MALWARE BY USING OUR APPROACH

	Ramnit	Lollipop	Kelihos_ver1	Kelihos_ver3	Vundo	Simda	Tracur	Obfuscator.ACY	Gatak
Ramnit	92.2	2.7	0.2	1.2	0.5	1.3	0.2	0.6	1.1
Lollipop	0.1	95.1	0.8	0.4	0.7	0.3	1.3	1	0.3
Kelihos_ver1	1.1	1.4	92.5	2.5	0.1	0.7	0.4	1.2	0.1
Kelihos_ver3	0.5	0.9	2.1	94.6	0.5	0.1	0.6	0.4	0.3
Vundo	0.4	1.8	0.6	0.3	95.6	0.6	0.4	0.1	0.2
Simda	0.2	0.1	0.4	0.2	0.8	97.2	0.3	0.4	0.4
Tracur	0.1	0.3	0.2	0.3	0.1	0.1	98.4	0.2	0.3
Obfuscator.ACY	1.9	1	0.1	0.1	0.1	0.3	1.7	92.7	2.1
Gatak	0.2	0.5	0.9	0.8	0.2	0.5	1.2	0.1	95.6

TABLE II. THE MATRIX OF THE CONFUSION IN MICROSOFT BIG 2015 FOR NINE TYPES OF THE MALWARE BY USING RESNET-50

	Ramnit	Lollipop	Kelihos_ver1	Kelihos_ver3	Vundo	Simda	Tracur	Obfuscator.ACY	Gatak
Ramnit	83.1	3.6	0.8	2.5	1.5	2.1	1.3	2.7	2.4
Lollipop	1.5	86.3	4.5	2.7	3.5	0.5	0.2	0.2	0.6
Kelihos_ver1	0.2	1.3	79.4	5.6	2.7	4.4	0.6	2.6	3.2
Kelihos_ver3	3.2	2.6	1.8	82.9	1.2	1.3	2.7	1.6	2.7
Vundo	0.3	0.1	0.5	0.2	96.8	0.6	0.1	0.5	0.9
Simda	0.3	0.6	0.7	0.1	0.6	96	0.4	0.7	0.6
Tracur	0.2	0.4	0.3	0.1	0.4	0.1	98.1	0.3	0.1
Obfuscator.ACY	3.1	0.1	0.3	0.7	2.3	1.2	0.7	89.9	1.8
Gatak	1	0.7	2.1	2.7	0.1	0.4	0.1	0.3	93.5

TABLE III. THE MATRIX OF THE CONFUSION IN MICROSOFT BIG 2015 FOR NINE TYPES OF THE MALWARE BY USING ALEXNET

	Ramnit	Lollipop	Kelihos_ver1	Kelihos_ver3	Vundo	Simda	Tracur	Obfuscator.ACY	Gatak
Ramnit	80.6	5.2	1.3	0.9	0.1	4.6	6.7	0.2	0.4
Lollipop	3.6	82.7	3.6	3.1	0.6	1.7	4.3	0.1	0.3
Kelihos_ver1	1.4	2.5	83.5	3.2	1.1	0.1	0.6	6.4	1.2
Kelihos_ver3	4.5	1.6	6.5	78.8	3.8	2.1	0.1	0.8	1.8
Vundo	0.6	0.7	1.2	6.5	80	3.2	5.6	0.7	1.5
Simda	0.5	0.1	0.2	0	1.1	97.3	0.5	0.1	0.2
Tracur	0.4	0.2	0.1	0.2	0.3	1.2	96.6	0.9	0.1
Obfuscator.ACY	1.6	2.6	0.5	1.3	0.2	0.1	2.8	87.5	3.4
Gatak	0.9	1.6	4.5	4.4	0.2	2.6	0.2	0.1	85.5

TABLE IV. THE COMPARISON OF OUR APPROACH WITH THE ADVANCED ALGORITHMS ON MALIMG

Method	Accuracy
Method in [25]	93.72
Method in [26]	94.50
Method in [27]	95.33
Method in [28]	96.08
Method in [29]	96.30
Proposed Method	98.12

TABLE V. THE COMPARISON OF OUR APPROACH WITH THE ADVANCED ALGORITHMS ON MICROSOFT BIG 2015

Method	Accuracy
Method in [25]	93.57
Method in [26]	93.40
Method in [27]	94.64
Method in [28]	94.24
Method in [29]	91.27
Proposed Method	95.36

D. Future Works

The proposed deep learning architecture shows some resistance to obfuscation, as evidenced by satisfactory results on the Microsoft BIG 2015 Dataset, which includes obfuscated malware samples. However, the method has not been tested against adversarial attacks with crafted inputs. Future research aims to evaluate the method's resilience to evasion attacks. Misclassification can occur due to similarities in features among different malware families. The model has only been tested on the Malimg and Microsoft BIG 2015 datasets. Future research could involve evaluating the model on additional datasets. The current architecture was implemented with limited computational power and resources. Plans for future work include deploying the model in a cloud computing environment to leverage greater computational power and resources. Additionally, future studies will use fewer hidden layers to reduce model complexity and will focus on extending the model by incorporating explainable AI and the latest feature optimization techniques to enhance real-time malware detection.

V. CONCLUSIONS AND SUGGESTIONS

The IoTs technology deepens the attendance of the connected sensors to the Internet in our daily behaviors, and brings the many benefits in the quality of the life. Also, it has generated the related problems to the security issues. Accordingly, the security solutions for the Internet of Things should be developed. In the current article, a new impressive cross-platform malware detection method based on artificial intelligence is designed for the industrial sensors of IoTs. In the presented approach, D3WT is used for the extraction of the feature. In addition, a combination from CNN and LSTM is used, to detect the malwares. Our approach is evaluated on Malimg and Microsoft BIG 2015. First, the proposed approach is compared with every network as separately. The obtained results confirm which our approach can impressively classify the malwares with the great values of the accuracy, the sensitivity, the specificity and the F1-score. When tested on the Microsoft BIG 2015 and Malimg datasets, the accuracy achieved is 95.36% and 98.12%, respectively. Then, the presented approach has been analyzed with the advanced models. The obtained outcomes show the benefit and the superiority of our approach against the similar models.

The model has only been tested on the Malimg and Microsoft BIG 2015 datasets. Future research could involve evaluating the model on additional datasets. The current architecture was implemented with limited computational power and resources. Plans for future work include deploying the model in a cloud computing environment to leverage greater computational power and resources. For further study, provision of a detection system that specifically classifies the malwares which uses the obfuscation techniques, can be considered. In addition, the next researches can rely on the deployment of the presented approach, by integrating the model basis on AI by the latest techniques for the optimization of the feature, to enhance the detection of the malware in the real time.

REFERENCES

[1] Miorandi, D., Sicari, S., De Pellegrini, F., Chlamtac, I., 2012. Internet of Things: vision, applications and research challenges. Ad Hoc Netw.10(7).1497-1516.

[2] Atzori, L. Iera, A., Morabiti, G., 2010. The internet of things: A survey, computer Network, V54, 15, 2787-2805.

[3] Borgia, E., 2014. The Internet of Things vision: key features, applications and open issues. Comput Commun.54, 1-31.

[4] Shigen Shen; Longjun Huang; Haiping Zhou; Shui Yu; En Fan; Qiying Cao, Multistage Signaling Game-Based Optimal Detection Strategies for Suppressing Malware Diffusion in Fog-Cloud-Based IoT Networks, IEEE Internet of Things Journal.

[5] Sicari, S., Rizzardi, A., Grieco, L., Coen-Porisini, A., 2015. Security, privacy and trust in Internet of Things: the road ahead. Comput. Netw. 76 (0), 146–164.

[6] Mudgerikar, A., Sharma, p., & Bertino, E.,2019. A system- level Intrusion Detection System for IoT Devices. In Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, 493-500.

[7] N. Dosti.,2019. New mechanism to enhance IoT network security using quantum and classical cryptography (in Persian), Journal of Electronical & Cyber Defence, Vol 4.

[8] Kumar, S., Dutta, K., 2016. Intrusion detection in mobile ad hoc networks: techniques, systems, and future challenges. Secur. Commun. Netw. 9 (14), 2484–2556.

[9] Midi, S., Krishna, P., Agarwal, H., Saxena, A., Obaidat, M., 2011. A learning automata based solution for preventing Distributed Denial of Service in Internet of Things. In: Internet of Things (iThings/CPSCom), International Conference on and Proceedings of the 4th International Conference on Cyber, Physical and Social Computing, 114–122.

[10] Butun, I., Morgera, S., Sankar, R., 2014. A survey of intrusion detection systems in wireless sensor networks. Commun. Surv. Tutor. IEEE 16 (1), 266–282.

[11] Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., Rajarajan, M., 2013. A survey of intrusion detection techniques in Cloud. J. Netw. Comput. Appl. 36 (1), 42–57.

[12] Mitchell, R., Chen, I.-R., 2014. A survey of intrusion detection techniques for cyberphysical systems. ACM Comput. Surv. (CSUR) 46 (4), 55.

[13] Granjal, J., Monteiro, E., Silva, J.S., 2012. On the effectiveness of end-to-end security for Internet-integrated sensing applications. In: Green Computing and Communications (GreenCom), IEEE, 87–93.

[14] Le, A., Loo, J., Chai, K.K., Aiash, M., 2016. A specification-based IDS for detecting attacks on RPL-based network topology. Information 7 (2), 25.

[15] Arwa Aldweesh, Abdelouahid Derhab., 2020. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. Knowledge-Based Systems. Vol 189.

[16] Klempous, Ryszard, et al., 2007. Adaptive misbehavior detection in wireless sensors network based on local community agreement. 14th Annual IEEE International Conference and Workshops on the Engineering of Computer- Based System.

[17] A. Marosi, E. Zabah, H. Ataee khabaz.,2020. Network intrusion detection using a combination of artificial neural networks in a hierarchical manner (in Persian), Journal of Electronical & Cyber Defence , Vol 8, pp. 89-99.

[18] Al-Timime ZS. Signal denoising using double density discrete wavelet transform. J Al-Nahrain Univ Sci 2017;20(4):125–9. https://doi.org/10.22401/ jnus.20.4.19.

[19] Qiao YL, Song CY. Double-density dual-tree wavelet transform based texture classifcation. In: IIH-MSP 2009 - 2009 5th Int. Conf. Intell. Inf. Hiding Multimed. Signal Process. 1; 2009. p. 1322–5. https://doi.org/10.1109/IIH-MSP.2009.148.

[20] Islam MZ, Islam MM, Asraf A. A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. Informatics Med Unlocked 2020;20:100412. https://doi.org/10.1016/j.imu.2020.100412.

[21] Shahzad F, Mannan A, Javed AR, Almadhor AS, Baker T, Al-Jumeily OBE D. Cloud-based multiclass anomaly detection and categorization using ensemble learning. J Cloud Comput 2022;11(1). https://doi.org/10.1186/s13677-022-00329-y.

[22] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, ‘‘Malware images: Visualization and automatic classification,’’ in Proc. 8th Int. Symp. Visualizat. Cyber Secur. (VizSec), 2011, pp. 1–7.

[23] Microsoft Malware Classification Challenge (Big 2015). Accessed: Apr. 20, 2021. [Online]. Available: https://www. kaggle.com/c/malwareclassification.

[24] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. A. Khan, ‘‘Performance analysis of machine learning algorithms in intrusion detection system: A review,’’ Procedia Comput. Sci., vol. 171, pp. 1251–1260, Jan. 2020.

[25] J.-S. Luo and D. C.-T. Lo, ‘‘Binary malware image classification using machine learning with local binary pattern,’’ in Proc. IEEE Int. Conf. Big Data (Big Data), Dec. 2017, pp. 4664–4667.

[26] Z. Cui, F. Xue, X. Cai, Y. Cao, G.-G. Wang, and J. Chen, ‘‘Detection of malicious code variants based on deep learning,’’ IEEE Trans. Ind. Informat., vol. 14, no. 7, pp. 3187–3196, Jul. 2018, doi: 10.1109/tii.2018.2822680.

[27] D. Gibert, ‘‘Convolutional neural networks for malware classification,’’ M.S. thesis, Univ. Rovira Virgili, Tarragona, Spain, Oct. 2016.

[28] A. Singh, A. Handa, N. Kumar, and S. K. Shukla, ‘‘Malware classification using image representation,’’ in Proc. Int. Symp. Cyber Secur. Cryptogr. Mach. Learn. Cham, Switzerland: Springer, Jun. 2019, pp. 75–92.

[29] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, ‘‘Robust intelligent malware detection using deep learning,’’ IEEE Access, vol. 7, pp. 46717–46738, 2019.