A Multi-layer Deep Learning Approach for Malware Classification in 5G-Enabled IIoT

Imran Ahmed, Marco Anisetti, Gwanggil Jeon*

***Abstract*— 5G is becoming the foundation for the Industrial Internet of Things (IIoT) enabling more effective low-latency integration of artificial intelligence and cloud computing in a framework of a smart and intelligent IIoT ecosystems enhancing the entire industrial procedure. However, it also increases the functional complexities of the underlying control system, and introduce new powerful attacks vectors leading to severe security and data privacy risks. Malware attacks are starting targeting weak but highly connected IoT devices showing the importance of security and privacy in this scenario. This paper designs a 5G-enabled system, consisted in a deep learning-based architecture aimed to classify malware attacks on the IIoT. Our methodology is based on an image representation of the malware and a Convolutional Neural Networks (CNNs) that is designed to differentiate various malware attacks. The proposed architecture extracts complementary discriminative features by combining multiple layers achieving 97% of accuracy.**

***Index Terms*— 5G, Cybersecurity, Deep learning, Industrial IoT, Malware detection**

I. Introduction

INTERNET of Things (IoT) allows both conventional electronics and daily 'things' embedded with sensors, computing, communication, and networking abilities to connect to the Internet in order to transmit and receive data. It has been used in many emerging applications, like smart cities and big data [1], and [2]. There is no doubt that the ability of typical IoT applications originates from their capabilities to gather, interpret, and communicate with a user's life in a pervasive and devoted fashion. IoT applications connect sensors and devices over multiple verticals, including healthcare, agriculture, manufacturing enterprise, business consumer and services, and other intelligent cities applications [3]. IoT-based systems include smart devices and sensors for such applications as home automation, monitoring, wearable sensors, TV, and mobile applications that usually do not produce emergency conditions if something goes wrong, as shown in Figure.1.

On the other hand, IIoT applications connect devices, sensors, actuators, and machines at the industrial level to enhance manufacturing and industrial processes such as robotics, automation, processing of heavy machinery, automobile and transportation, health care, utilities, and manufacturing. In IIoT deployments, operation failures and downtime can occur in high-risk situations or indeed cause life-threatening ones. Moreover, the applications are concerned with increasing efficiency and enhancing health or safety versus IoT applications for user-centric nature. The difference between IoT and IIoT can be depicted in Figure.1. Although IoT and IIoT have various technologies in general, including cloud platforms, internet connections, cellular networks, sensors, connectivity, machine-to-machine interactions, and data analytics techniques, utilized for different purposes.

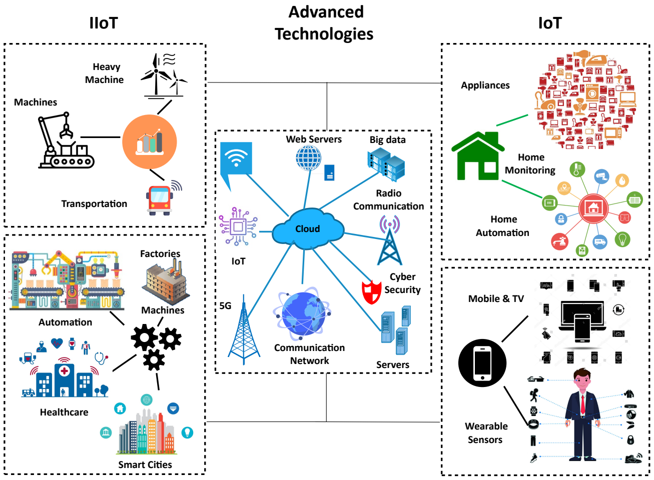

Fig. 1: General illustration of Industrial IoT and IoT with advanced technologies. Both have different applications in terms of services, requirements, and constraints.

Industry 4.0 and the IIoT are being implemented across various industries, vertical businesses converging automotive, customer assets, services, pharmaceuticals, food and beverage, manufacturing, and several others in the modern era [4]. Wireless connectivity remains a vital component of this development, giving pervasive and powerful connections across devices, machines, systems, people, and objects. 5G is also poised to influence automated manufacturing transformation, especially individual on-premise, and public 5G solutions. This signifies one of the most important opportunities to expand next-generation wireless communications [5]. It has been mainly devised to achieve high-speed data throughput with low latency. The development of 5G changes the application of IIoT systems in two main ways. First, the IIoT is the convergence of intelligent technology that allows machines to solve critical industry problems and increase operational

Imran Ahmed, School of Computing and Information Science, Anglia Ruskin University Cambridge, East Road, Cambridge, CB1 1PT, UK. Email:imran.ahmed@aru.ac.uk.

Marco Anisetti, Department of Computer Science, Università degli Studi di Milano, Celoria 18, Milano, Itlay. Email: marco.anisetti@unimi.it.

Gwanggil Jeon*, Department of Embedded Systems Engineering, Incheon National University, Incheon, Korea. E-mail: gjeon@inu.ac.kr.

performance [6]. Second, IIoT is similar to an industrial machine. Like IoT, it concentrates on the interconnection of IoT sensors, devices, machinery/equipment, and the software that powers the particular industrial application process. By integrating hardware, software, data collection, and advanced data analytics techniques, such as predictive and prescriptive analysis, advanced systems can develop by leveraging tools and real-time insights on industry performance.

Furthermore, advanced artificial intelligence solutions enable deeper insights and more intelligent, more agile systems that make an operational performance at an industrial scale [7]. Going forward, IIoT plays a significant part in digital transformations, particularly to digitize production lines, manufacturing processes, and supply chains. IIoT holds great potential in manufacturing [8], especially for quality checks, green and sustainable applications, traceability of the supply chain, and overall efficiency. It is the key to processes for predictive maintenance [9], where it improves field service, energy control, and asset tracking. It also advanced the automotive and transportation industry, which widely applies industrial robots and smart IIoT devices in manufacturing processes and systems. The IIoT helps to proactively manage these systems and point out possible difficulties before they can interrupt production. Industrial sensors are also used in agriculture [10] to collect soil nutrients, moisture, and other related data, enabling farmers and yielders to produce an optimal yield. The oil and gas enterprise.also applies industrial IoT devices and systems to support a line of independent aircraft that can perform thermal and visual imaging to identify possible pipeline difficulties. This information is coupled with data from different kinds of sensors to assure secure operations. Another important area that uses IIoT is the health sector to collect data on patients and diseases and provide essential medical facilities.

With a lot of advantages, the biggest risk associated with IIoT application is a concern to security [11]. Even after being placed into operation, it is normal for IIoT devices, systems, and machines to continue applying default passwords. Furthermore, several IIoT devices and sensors transfer data as plain text; these situations can make it comparatively easy for an intruder and attacker to intercept the data coming or generating from an IIoT device or systems [12]. Likewise, an intruder can take over an insecure and vulnerable IIoT machine or system and apply it as a platform for originating attacks or threats against other network systems and resources. Thus, security is a big hurdle for those who are liable for the coordination of IIoT systems, but so, moreover, is system control. As an industry adopts more and more IIoT based systems, it will become increasingly necessary to use an efficient system control strategy. Assurance has been identified as a suitable way to control a target system against security concerns [13], computing, for instance, a risk indicator [14]. More particularly, the industry must be capable of confidently identifying IIoT systems to prevent the use of miscreant or unreliable devices and systems. Building a means of recognizing each particular sensor, device, and system is also essential to replace a lost sensor or device or make a device or sensor update system.

In this work, we used artificial intelligence and developed a deep learning-based malware detection system for a 5G enabled IIoT system. The developed system is based on multi-layer CNN architecture for the classification of various types of malware attacks. The proposed architecture integrates an adequate number of layers, trained and tested on benchmark data set. Further, the system is 5G enabled, thus, providing high throughput and low latency and can make it feasible for sensors and devices to share data in real-time when deployed on a 5G data-intensive solution such as the one in [15]. This makes the system more efficient than previous ones, in which real-time connectivity is only possible when the devices are located on private networks with high-speed connectivity. Therefore, the developed system can support real-time connectivity applications such as autonomous vehicles and other innovative city applications. The primary contribution of the paper is provided as follows:

- To apply artificial intelligence and develop a multi-layer CNN-based architecture for malware attacks classification in IIoT.
- To apply data pre-processing techniques and transfer learning to enhance the performance of CNN architectures.
- To explore training and testing observations of the CNN model using benchmark data set.
- To investigate and compare the results of the developed system with different CNN based architecture for malware classification in terms of accuracy.

The rest of the work performed in the paper is categorized in the following sections: In Section II, a summary of related works is presented that is used for malware classification in various IIoT applications. In Section III, we present a 5G-enabled system for IIoT, which is based on artificial intelligence. We also explained a deep learning-based real-time system and CNN architecture used for multiple malware attack classification. In Section IV, the summary of the data set used for the experiments is briefly explained. Furthermore, this section discussed the testing and performance results using different evaluation matrices. Lastly, in Section V, we concluded the presented work with possible future directions.

II. Related Work

In recent years, many efforts have been performed to assure the security of IoT-based systems, sensors, and devices. The efforts range from developing secured IoT systems to devising prevention and detection mechanisms of malicious activities in IoT-based applications.

Authors in [16] introduced a technique for an IoT-based cyber-physical system that identified the difference in the expected behavior of network links in order to identify the opponent nodes. Pajouh et al. [17] developed an intrusion detection system using machine learning to identify various kinds of attacks in an IoT infrastructure. The proposed method classified several attacks utilizing the Naive based and K-Nearest Neighbor methods after decreasing the extracted features using the principal component algorithm and linear discriminate analysis. In [18], authors introduced an intrusion

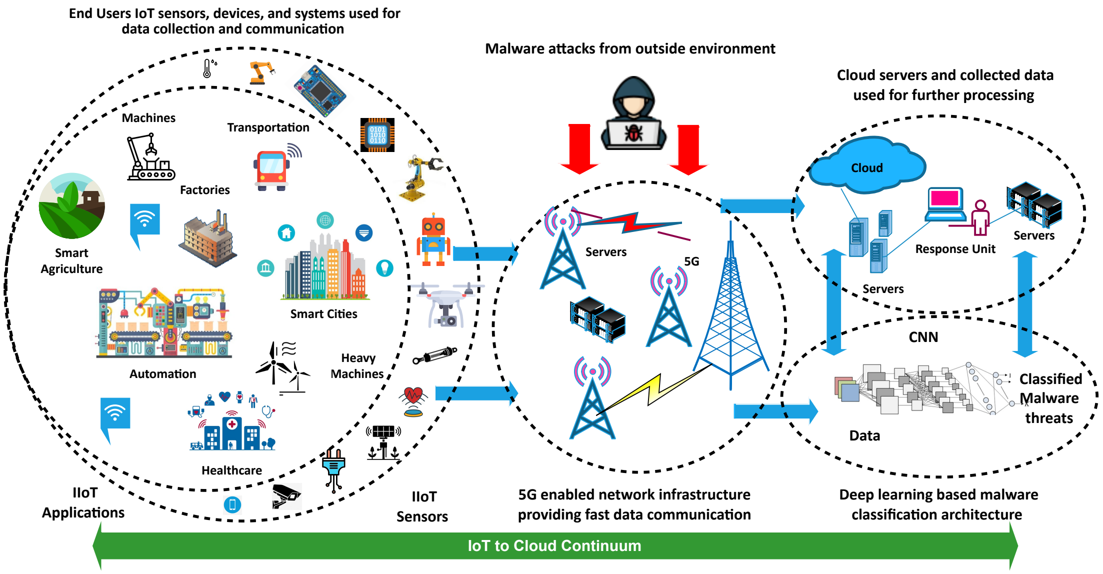
End Users IoT sensors, devices, and systems used for data collection and communication
Malware attacks from outside environment
Cloud servers and collected data used for further processing
Machines
Transportation
Factories
Smart Agriculture
Smart Cities
Automation
Heavy Machines
Healthcare
Servers
5G
Cloud
Response Unit
Servers
Servers
CNN
Classified Malware threats
Data
IIoT Applications
IIoT Sensors
5G enabled network infrastructure providing fast data communication
Deep learning based malware classification architecture
IoT to Cloud Continuum

Fig. 2: The overall Methodology for 5G-enabled malware detection on IIoT.

detection method for IoT based systems that detect denial of service attacks using packet transmission rate. Li et al. [19] performed intrusion detection to decrease several malicious threats in an IoT by using the mechanism of blockchain. Authors in [20] presented an IoT-based security system to identify the anomalies directly by using a kernel of the Linux operating system. In [21], introduced an IoT-based security system that worked in two stages. In the initial stage, the system mapped the identities and names with the file produced during the training process, and in another stage, training examples were given to separate the rival nodes from the reliable ones. The most valuable section of the suggested scheme was the performance of the machine learning method at the connection levels. Ahmed et al. [22] introduced an identification scheme that also repeatedly used two stages process. An artificial neural network design was adopted to detect anomalies in the first stage. A tag method was added during the other stage, which was applied to identify the malicious connections.

Wazid et al. [23] presented an edge IoT system for intrusion detection to protect from routing attacks. Zhao et al. [24] introduced several communication applications that added software-based networking ideas using machine learning methods. Authors principally concentrated on the importance of traditional machine learning algorithms in SDN-based networks. Zhao and Dong [25] suggested a feature based selection method using potential entropy evaluation criteria to examine the number of the data by weighing their importance. Authors in [26] suggested a security threat identification system using machine learning techniques to recognize and describe intrusions. The generated IoT data transmitted from end devices to the data server was further processed by machine learning that helped in the detection of malware attacks. Similarly, a lot of other efforts are also made by different researchers for malware detection using different data sets and samples. Like Aziz et al. [27], applied feature extraction method and feed-forward neural network for classification of malware threats utilizing a data set of 1710 samples obtained from eight malware classes. Authors in, [28] used heterogeneous global and local characteristics or features of malware images and then used machine learning to classify threats. Their work used 9339 samples from 25 different classes of malware. Mahmoud et al. [29] developed CNN architecture to analyze malware attacks from the binary executable corpus. Furthermore, this architecture randomly chooses 10% samples to examine the malware classes in every sequence. Zhihua et al. [30] produced a malware classification paradigm applying CNN on 9339 samples obtained from 25 distinct malware classes.

From the above brief discussion, it is concluded that researchers to detect, identify, and classify malicious activities and malware attacks and threats. In addition, researchers used various data sets with varying no of samples for training and testing experimentation's, but mostly utilized a limited number of data sets and classes of malware attacks. However, in this work we proposed a deep learning-based 5G enabled malware classification system for Industrial IoT. Our presented system comprises of multi-layer CNN architecture that can classify 25 different classes of attacks and achieves good results on benchmark data sets.

III. 5G-Enabled Malware Detection in IIoT

In this work, we presented a 5G enabled IIoT system; the developed system can be used for various applications, e.g., smart agriculture, manufacturing, healthcare, transportation, and smart cities. The developed system will also be capable of operating various machines in different industries. It can be seen that various intelligent sensors, actuators, cameras, robots, machines, IC controllers, IoT based chips collect the necessary information and communicate that information using the internet and 5G infrastructure. The 5G infrastructure provides fast communication links and a low latency rate to the smart sensors, devices, and systems to efficiently process the information to the cloud servers. However, like conventional methods, communication links and IoT devices, sensors, and systems are also targeted by intruders and attackers. The activities of intruders intercept the data coming or generating from an IIoT device or system. An intruder can take over an insecure and vulnerable IIoT machine/device or system and apply it as a platform for originating attacks or threats against other network systems and resources, which affect the normal operations of the system within industries. Thus, a smart system is needed here that can identify such attacks and prevent the systems from unnecessary operations.

In this paper, we adopted the 5G-enabled solution proposed in [5] to securely collect information about the IIoT system in operation with the scope of using them to detect malware activities. In order to classify malware attacks from the collected information, we leverage artificial intelligence and pass the information through a multi-layer CNN architecture. As an industry adopts more and more IIoT based systems, it will become increasingly necessary to use an efficient system and malware control strategy. More particularly, the industry must be capable of confidently identifying IIoT systems to prevent the use of miscreant or unreliable devices and systems. As from Figure 2, it can be seen that the collected information is passed through a deep CNN architecture that can help to classify the malware attacks, and the information is sent back to the data servers connected to the cloud. The response unit can perform the necessary action and prevent IIoT applications from attackers and unusual activities. The number of samples of data set containing 25 different classes of malware attacks.Moreover, the details of the developed deep learning architecture used for the classification of malware attacks are provided in Figure 3. It can be seen that the developed system is divided into two main sections, visualization of malware attacks into grayscale images, which is done using pre-processing, and the proposed multi-layer CNN architecture. The effectiveness of the CNN architecture in malware image classification/identification/recognition tasks is the primary purpose of using the proposed technique. The detail of the proposed system is provided in the following subsections:

A. Data Pre-Processing

The CNN architecture developed in this work has experimented on the Malimg data set [1], consisting of 9,339 malware samples from 25 distinct malware classes [31]. As shown in Figure 3, malware binaries are converted into an 8-bit unsigned integer that is composed of a matrix $M \epsilon Rm \times n$. The obtained matrix is further visualized as a grayscale image possessing values ranging between [0 to 255], where 0 expresses black and 1 represents white. The converted binary images are shown in Figure 3. The obtained grayscale images are resized to a 2-dimensional matrix of 64 × 64 and are flattened into an $n \times n$ size array. Every feature array is further labeled with its identical indexed malware class name (i.e., 0 - 24). Later, the features are normalized using Equation 1.

$$z = \frac{X - \mu}{\sigma}. \tag{1}$$

where X is the feature that is normalized, $\sigma$is its standard deviation, and μ is its mean value. The data set we used is unbalanced; to balance the data set, several methods are available in the literature; however, in this work, we used the class weight method in which higher weight is assigned to the minority class, and lower weight is assigned to the majority class. In this way, weights values of y axis are automatically adjusted and inversely proportional to the frequencies of the corresponding class in the input data.

B. CNN Architecture for Malware Classification

After converting malware attacks into grayscale images, the data set is split into three categories, (i) training data samples, (ii) testing data samples, and (iii) validation data samples. For training of CNN architecture, we used 80% of the data. The multilayer CNN architecture is used to learn the features as shown in Figure 3. The input and the predicted outcome are involved in the training process. The proposed architecture consists of pre-processed input images, multiple CNN layers, Max-pooling layer, Flatten layer, and a fully connected layer used for classification. As stated earlier, that input image is an array of pixels. A two-dimensional grayscale input image matrix is produced, which is further given to the multilayer CNN architecture, which will make further processing on the malware images. The overall architecture consists of multiple Convolution layers, Maxpooling layers followed by each Convolution layer, the Dense layer, the activation layer, and the fully connected layer.

The CNN layers obtained a feature map, for which an operation of convolutional is made on the matrix as an input image. The extracted features of the CNN layers are reduced using the max-pooling layer. It reduces filter sensitivity, variations, and noise. All convolution layers work similarly to the first; however, the primary layer accumulates low level characteristics or features from the input image, while other layers mainly extract the high-level features. After that, in other max-pooling layers followed by convolutional layers, the exact function of the first max-pooling layer is offered for reducing the feature map dimensionality. It produces an array of feature pools. The flattening layer performs a method on the matrix, which is received from the last max-pooling layer. A pooled feature matrix is then transformed into a feature vector by this layer, and a column or vector that is obtained. The obtained feature vector from the flatten layer is utilized for classification. In this work, we applied the softmax activation functions for classification. Finally, the input images are examined for malware classification.

As shown in Figure 3, the input image consists of an array of pixels. A two-dimensional image matrix at the input is produced by applying a shape parameter given to CNN architecture for further processing. The primary layer in a CNN architecture performs the feature extraction. The matrix

[1] https://vision.ece.ucsb.edu/research/signal-processing-malware-analysis

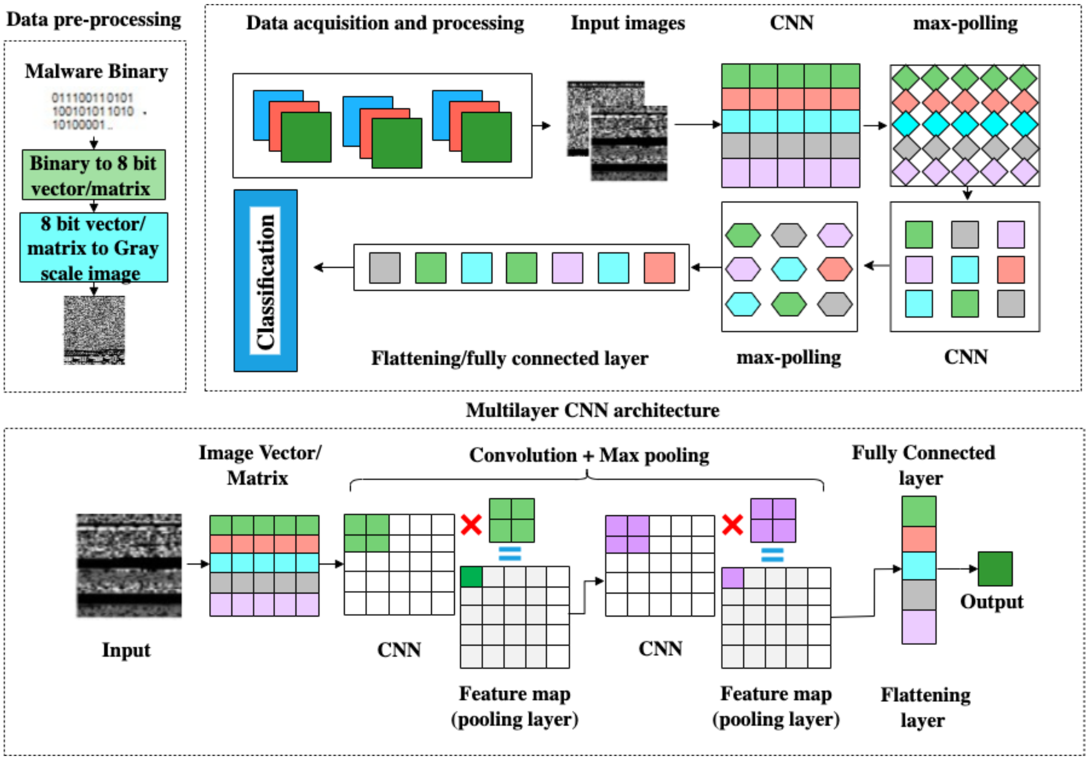
Data pre-processing
Malware Binary
011100110101
10010101 1010 .
10100001..
Binary to 8 bit vector/matrix
8 bit vector/ matrix to Gray scale image
Data acquisition and processing
Input images
CNN
max-polling
Classification
Flattening/fully connected layer
max-polling
CNN
Multilayer CNN architecture
Image Vector/ Matrix
Convolution + Max pooling
Fully Connected layer
Input
CNN
Feature map (pooling layer)
CNN
Feature map (pooling layer)
Flattening layer
Output

Fig. 3: A brief technical overview of the proposed system, (a) Visualizing malware as a gray scale image. (b) Proposed CNN architecture, (c) Detailed of the proposed CNN architecture.

of features is generated by convolving the input image matrix with training filters, as shown in Figure 3. A filter matrix K is employed for convolution on image matrix I, producing a function map F. Equation 2, is utilized to estimate the matrix F, as follows:

$$F_{r,c} = R\big(K_{w,b} \circledast I_{r+w-1,c+h-1} + b\big). \tag{2}$$

In Equation 2, the matrix row is indicted with r, the matrix column is represented with c, the width and height of the filter are described with w and h, and I represents the limit of the filter. The value of r is ranging between $1 \leq w \geq l$; the variable c varies from $1 \leq h \geq l$. The activation function (ReLU) is represented with R. $\circledast$ indicating a convolutional function, and b holds the bias value. The primary goal of the ReLU is to represent the non-linearity in the architecture, applying $f(x) = max(x, 0)$. To calculate the matrix F, we apply the below equation:

$$F = [f_{1,1}, f_{1,2}, \cdots f_{1,n}]. \tag{3}$$

After each convolution layer, the sub-sampling layer, also recognized as the max-pooling layer, reduces extracted input function maps, also called down-sampling. In this work, we perform the operation of max-pooling on the feature map in order to reduce its size and dimensions. The given equation is applied to estimate the max-pooling function.

$$M_{r,c} = max(F_{r+w-1,c+h-1}). \tag{4}$$

While the feature map at pooling layer is estimated as follows:

$$M = [m_{1,1}, m_{1,2}, \cdots, m_{n,m}]. \tag{5}$$

The other convolution layers are applied to obtain high-level characteristics/features from the max-pooling layer input. The computation time of each convolution layer is the same as the initial convolution layer (Equation 2, and Equation 3). The purpose of the other max-pooling layers is to overcome the dimension of the feature matrix. Other layers are also computed similarly (Equation 4, and Equation 5). This layer is at the end of the max-pooling layer, which takes input from the final max-pool layer. The principal objective of a max-pooling layer is to transform a feature vector or column from a feature matrix obtained from the pooling layer. The feature map M components are re-shaped into the vector of features by re-structuring the function, defined as:

$$F_v = pooled.reshape(f - w + 11(v - h + 1)). \tag{6}$$

At the end, for classification, the probabilities for the different malware classes are determined by shaping a dense layer with a fully connected layer with multiple neurons applying softmax functions. Therefore, the net input is obtained as follows:

$$u_j = \sum_{i}^{f} w_i x_i + b. \tag{7}$$

In Equation 7, weight vector is represented with w, input vector with x, and bias term is represented with b. The above discussed classification architecture classifies 25 different classes of malware attacks for industrial IoT environments.

IV. Experimental Results

This section provides a detailed discussion of the various experiments performed to assess the performance of the above-discussed system. A publicly available data set, namely, the Malimg data set, was used for the experimentation of the proposed malware detection method. As stated previously, the Malimg data set comprised 9339 samples of 25 malware classes. The experiments were performed on an Intel Core M3, 7th Generation, 64-bit operating system with 8GB RAM; the proposed CNN model was introduced using python programming language, Keras, and TensorFlow libraries. A flexible environment is designed for deep learning models adopting hyperparameters. It does operations by applying multi-dimensional arrays to achieve parallel execution and speed up the classification process. The training data set is comprised of 70% of the data, while for testing, 20% of the original data is used to assess the CNN architecture. In addition, data validation is used to overcome underfitting and overfitting, which happens when training performance is significant, and it decreases when the architecture is tested on the new data. The training and testing observations are shown in Figure 4 and 5. The training and testing loss shows that it can be seen that both loss curves are descending values; the training and testing values range between 0.8 and 0.5. Similarly, as seen in Figure 5, the training and testing accuracies are increasing with a number of epochs and improving the performance of the model. The training and testing accuracies range between 0.8 and 0.85.

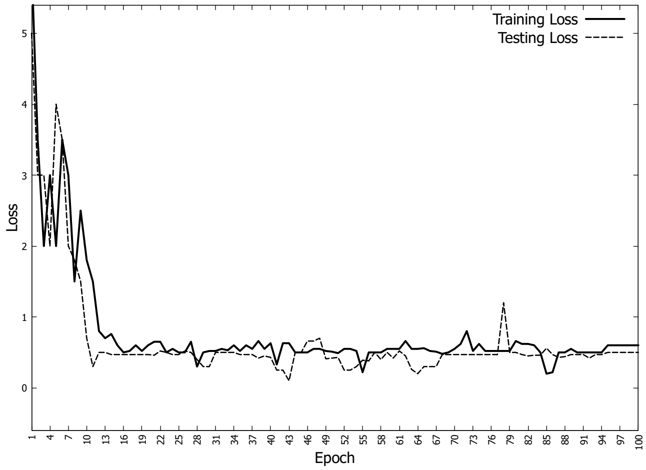

Fig. 4: Training and Testing Loss.

In Figure 6, we have shown the classification results of the above-discussed architecture. The above models effectively classify 25 different attacks of malware families. It can be seen that all malware attacks in the gray-scale images are more alike each other. However, still, the proposed deep learning model effectively classifies the images with excellent results. The efficiency of the model is evaluated using different evaluation parameters. The evaluation parameters are calculated using a confusion matrix where we note good performances for most of the malware apart from Autorun.K that is constantly mistaken for Yuner.A. and Swizzor.gen!E that in many cases was mistaken as Swizzor.gen!l.

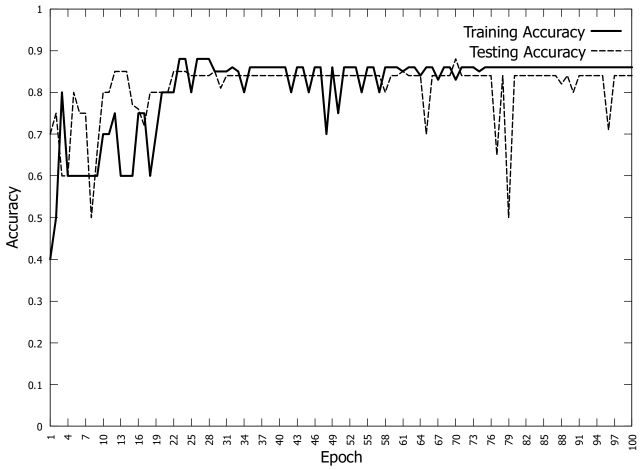

Fig. 5: Training and Testing Accuracy.

We determined Accuracy, Precision, and Recall for all different kinds of malware attacks as depicted in Table I. It can be seen that the architecture achieves maximum results for Agent.FYI, Adialer.C, Allaple.L, Autorun.K, Dontovo.A, Dialplatform.B, Fakerean, Lolyda.AA3, Instantaccess, Malex.gen!J, Rbot!gen, Obfuscator.AD, VB.AT, Skintrim.N, and Yuner.A, that is 96%. The minimum accuracy is obtained for Swizzor.gen!I that is 65%.

TABLE I: Results of the model for different malware attacks.

S.No	Malware Class	Acc(%)	Prec(%)	Rec (%)
1	Adialer.C	96	94	98
2	Agent.FYI	96	94	97
3	Allaple.A	94	90	94
4	Allaple.L	96	93	97
5	Alueron.gen!J	92	93	95
6	Autorun.K	96	94	97
7	C2LOP.P	86	80	86
8	C2LOP.gen!g	80	79	82
9	Dialplatform.B	96	93	98
10	Dontovo.A	96	93	98
11	Fakerean	96	93	98
12	Instantaccess	96	93	98
13	Lolyda.AA1	93	92	92
14	Lolyda.AA2	92	90	92
15	Lolyda.AA3	96	92	98
16	Lolyda.AT	92	90	94
17	Malex.gen!J	96	90	97
18	Obfuscator.AD	96	92	97
19	Rbot!gen	96	92	97
20	Skintrim.N	96	92	97
21	Swizzor.gen!E	72	69	76
22	Swizzor.gen!I	65	62	69
23	VB.AT	96	93	97
24	Wintrim.BX	92	92	95
25	Yuner.A	96	94	97
Average		91.92	89.16	93.44

The average precision and recall value of the above discussed model for each malware class is presented in Figure 7. In the figure, the precision and recall percentage are plotted from 0 to 100. It can be seen that the architecture achieves excellent results, as the precision rate mostly classes, e.g., for Adialer.C, Agent.FYI, Allaple.L, Autorun.K, Dialplat-

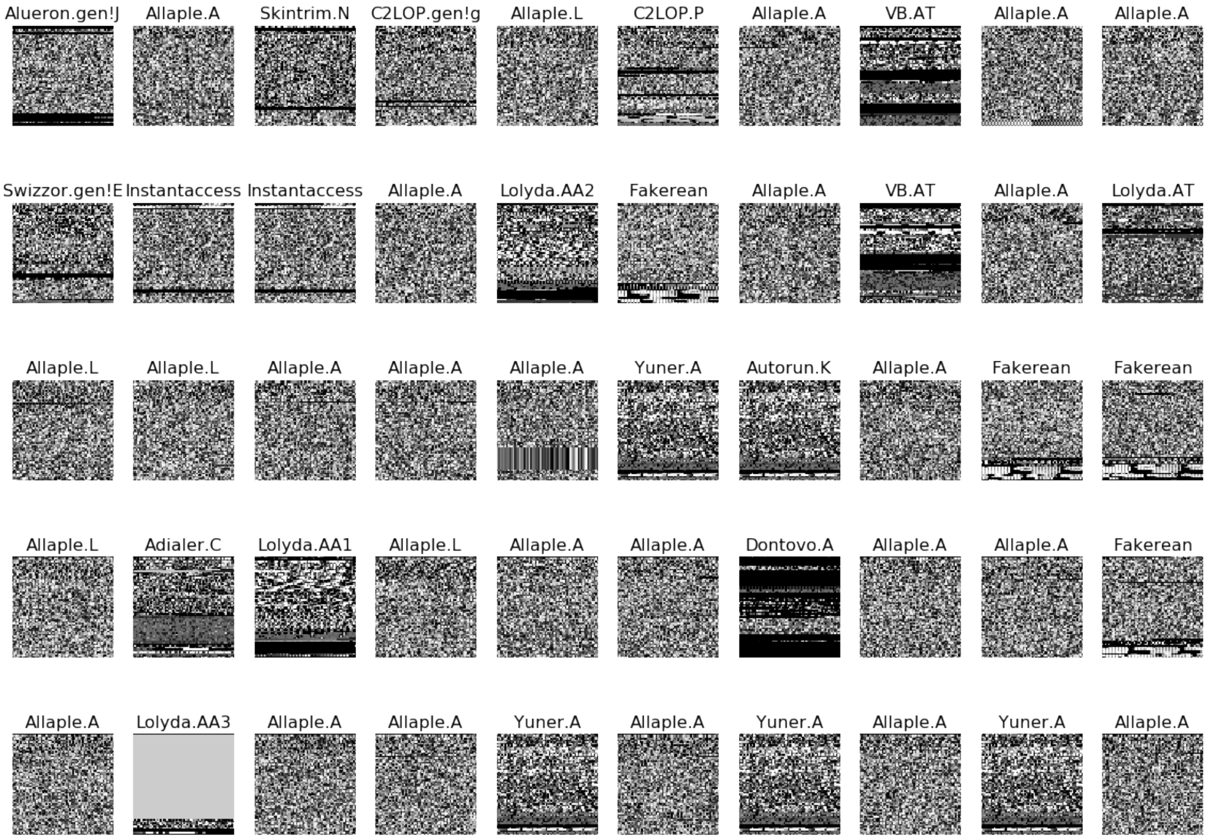

Fig. 6: Classification results of 25 different classes of malware attacks.

form.B, Dontovo.A, Fakerean, Instantaccess, Lolyda.AA3, Malex.gen!J, Obfuscator.AD, Rbot!gen, Skintrim.N, VB.AT, and Yuner.A, is more than 95%. The minimum result is obtained for Swizzor.gen!I that is 60%.

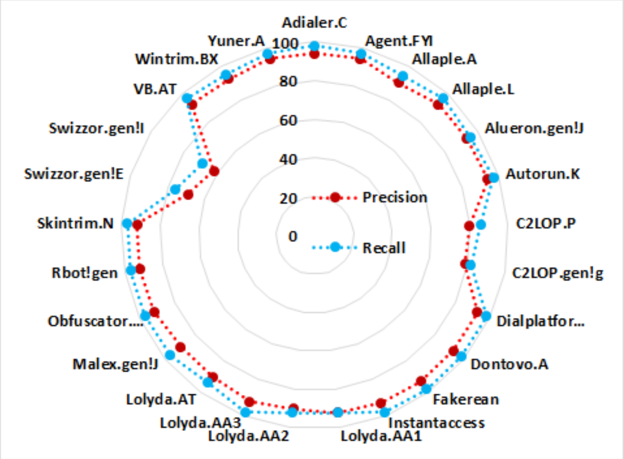

Fig. 7: Precision and Recall rate of the CNN architecture for different malware attacks.

The average accuracy rate with standard error is depicted in Figure 8; for all classes, the CNN architecture achieves good results for 25 different classes of malware attacks. The accuracy of Agent.FYI, Allaple.A, Allaple.L, Alueron.gen!J and Autorun.K, Dialplatform.B, Dontovo.A, Instantaccess, Lolyda.AA1, Lolyda.AA2, Lolyda.AA3, Lolyda.AT, Malex.gen!J, Obfuscator.AD, Rbot!gen, Skintrim.N, VB.AT, Wintrim.BX, and Yuner.A is more than 90%, while C2LOP.P, and C2LOP.gen!g is 86%. The minimum accuracy is recorded for Swizzor.gen!E, and Swizzor.gen!I.

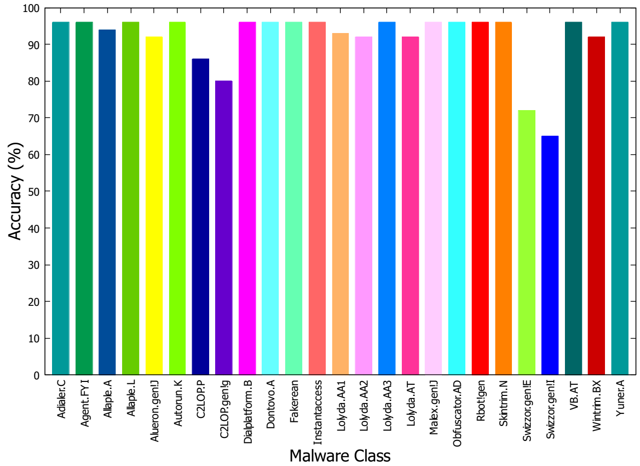

Fig. 8: Accuracy of the CNN architecture for different malware classes.

The comparison of the proposed method with traditional models is shown in Table II. The overall average accuracy of machine learning based models, including Support Vector

Machine (SVM) is 90%, Multilevel Perceptron is 92%, and Random Forest is 93%. In comparison, the deep learning based model like, VGG achieves 95% accuracy. It can be seen that among all the proposed CNN architecture gives good results, with an accuracy of 97%.

TABLE II: Comparison with traditional models.

S.No	Classification Algorithm	Acc (%)
1	SVM	90
2	Multilevel Perceptron	92
3	Random Forest	93
4	CNN VGG	95
5	Proposed CNN	97

V. Conclusions

This paper presented a 5G enabled system consisted deep learning-based architecture to classify malware attacks on the IIoT. A methodology is proposed for malware analysis is based on grays scale image visualization and a deep learning network. An integrated method is applied to propose a CNNs architecture that is designed to differentiate various malware attacks. The proposed architecture extracts complementary discriminative features by combining multiple layers. The system results are compared to previous methods; the experimental results reveal that the presented system's accuracy is improved. The presented system achieves 97% accuracy on the benchmark data set. In the future, we might continue this work for other types of cybercrime activities in other applications. We can also apply and utilize other deep learning-based models and architectures for the analysis, detection, and classification of different malicious activities.

References

[1] I. Ahmed, M. Ahmad, A. Ahmad, and G. Jeon, "Iot-based crowd monitoring system: Using ssd with transfer learning," *Computers & Electrical Engineering*, vol. 93, p. 107226, 2021.

[2] I. Ahmed, M. Ahmad, J. J. Rodrigues, and G. Jeon, "Edge computing-based person detection system for top view surveillance: Using centernet with transfer learning," *Applied Soft Computing*, vol. 107, p. 107489, 2021.

[3] M. Ahmad, I. Ahmed, and G. Jeon, "An iot-enabled real-time overhead view person detection system based on cascade-rcnn and transfer learning," *Journal of Real-Time Image Processing*, pp. 1–11, 2021.

[4] H. Jaidka, N. Sharma, and R. Singh, "Evolution of iot to iiot: Applications & challenges," in *Proc. of ICICC*, 2020.

[5] A. Mahmood, L. Beltramelli, S. F. Abedin, S. Zeb, N. Mowla, S. A. Hassan, E. Sisinni, and M. Gidlund, "Industrial iot in 5g-and-beyond networks: Vision, architecture, and design trends," *IEEE Transactions on Industrial Informatics*, 2021.

[6] P. Varga, J. Peto, A. Franko, D. Balla, D. Haja, F. Janky, G. Soos, D. Ficzere, M. Maliosz, and L. Toka, "5g support for industrial iot applications—challenges, solutions, and research gaps," *Sensors*, vol. 20, no. 3, p. 828, 2020.

[7] P. Trakadas, P. Simoens, P. Gkonis, L. Sarakis, A. Angelopoulos, A. P. Ramallo-González, A. Skarmeta, C. Trochoutsos, D. Calvo, T. Pariente *et al.*, "An artificial intelligence-based collaboration approach in industrial iot manufacturing: Key concepts, architectural extensions and potential applications," *Sensors*, vol. 20, no. 19, p. 5480, 2020.

[8] P. Deflorin, M. Scherrer, and K. Schillo, "The influence of iiot on manufacturing network coordination," *Journal of Manufacturing Technology Management*, 2021.

[9] M. Compare, P. Baraldi, and E. Zio, "Challenges to iot-enabled predictive maintenance for industry 4.0," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4585–4597, 2019.

[10] B. Almadani and S. M. Mostafa, "Iiot based multimodal communication model for agriculture and agro-industries," *IEEE Access*, vol. 9, pp. 10 070–10 088, 2021.

[11] C. A. Ardagna, R. Asal, E. Damiani, N. El Ioini, and C. Pahl, "Trustworthy iot: An evidence collection approach based on smart contracts," in *Proc. of IEEE SCC*. IEEE, 2019, pp. 46–50.

[12] J. Sengupta, S. Ruj, and S. D. Bit, "A comprehensive survey on attacks, security issues and blockchain solutions for iot and iiot," *Journal of Network and Computer Applications*, vol. 149, p. 102481, 2020.

[13] M. Anisetti, C. A. Ardagna, N. Bena, and E. Damiani, "An assurance framework and process for hybrid systems," in *International Conference on E-Business and Telecommunications*. Springer, 2020, pp. 79–101.

[14] M. Anisetti, C. A. Ardagna, N. Bena, and A. Foppiani, "An assurance-based risk management framework for distributed systems," in *Proc. of IEEE ICWS*. IEEE, 2021, pp. 482–492.

[15] M. Anisetti, F. Berto, and M. Banzi, "Orchestration of data-intensive pipeline in 5g-enabled edge continuum," in *Proc. of IEEE Edge*, 2022 (to appear).

[16] V. Sharma, I. You, K. Yim, R. Chen, and J.-H. Cho, "Briot: Behavior rule specification-based misbehavior detection for iot-embedded cyber-physical systems," *IEEE Access*, vol. 7, pp. 118 556–118 580, 2019.

[17] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K.-K. R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in iot backbone networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 2, pp. 314–323, 2016.

[18] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "Toward a lightweight intrusion detection system for the internet of things," *IEEE Access*, vol. 7, pp. 42 450–42 471, 2019.

[19] W. Li, S. Tug, W. Meng, and Y. Wang, "Designing collaborative blockchained signature-based intrusion detection in iot environments," *Future Generation Computer Systems*, vol. 96, pp. 481–489, 2019.

[20] D. Breitenbacher, I. Homoliak, Y. L. Aung, N. O. Tippenhauer, and Y. Elovici, "Hades-iot: A practical host-based anomaly detection system for iot devices," in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, 2019, pp. 479–484.

[21] A. Mudgerikar, P. Sharma, and E. Bertino, "E-spion: A system-level intrusion detection system for iot devices," in *Proc. of the ACM Asia conference on computer and communications security*, 2019, pp. 493–500.

[22] A. Saeed, A. Ahmadinia, A. Javed, and H. Larijani, "Intelligent intrusion detection in low-power iots," *ACM Trans. Internet Technol.*, vol. 16, no. 4, dec 2016.

[23] M. Wazid, P. Reshma Dsouza, A. K. Das, V. Bhat K, N. Kumar, and J. J. Rodrigues, "Rad-ei: a routing attack detection scheme for edge-based internet of things environment," *International Journal of Communication Systems*, vol. 32, no. 15, p. e4024, 2019.

[24] Y. Zhao, Y. Li, X. Zhang, G. Geng, W. Zhang, and Y. Sun, "A survey of networking applications applying the software defined networking concept based on machine learning," *IEEE Access*, vol. 7, pp. 95 397–95 417, 2019.

[25] L. Zhao and X. Dong, "An industrial internet of things feature selection method based on potential entropy evaluation criteria," *IEEE Access*, vol. 6, pp. 4608–4617, 2018.

[26] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for iot security based on learning techniques," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671–2701, 2019.

[27] A. Makandar and A. Patrot, "Malware class recognition using image processing techniques," in *Proc. of ICDMAI*. IEEE, 2017, pp. 76–80.

[28] H. Naeem, B. Guo, F. Ullah, and M. R. Naeem, "A cross-platform malware variant classification based on image representation," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 13, no. 7, pp. 3756–3777, 2019.

[29] M. Kalash, M. Rochan, N. Mohammed, N. D. Bruce, Y. Wang, and F. Iqbal, "Malware classification with deep convolutional neural networks," in *Proc. of 9th IFIP NTMS*. IEEE, 2018, pp. 1–5.

[30] Z. Cui, F. Xue, X. Cai, Y. Cao, G.-g. Wang, and J. Chen, "Detection of malicious code variants based on deep learning," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3187–3196, 2018.

[31] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: visualization and automatic classification," in *Proceedings of the 8th international symposium on visualization for cyber security*, 2011, pp. 1–7.