IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 24 February 2024, accepted 9 March 2024, date of publication 1 April 2024, date of current version 9 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3383831

RESEARCH ARTICLE

Deep Neural Networks for Enhanced Security: Detecting Metamorphic Malware in IoT Devices

FAIZA HABIB[1], SYED HAMAD SHIRAZI[2], KHURSHEED AURANGZEB[3], (Senior Member, IEEE), ASFANDYAR KHAN[2], BHARAT BHUSHAN[4], (Senior Member, IEEE), AND MUSAED ALHUSSEIN[3]

[1]Department of Computer Science, Abasyn University, Islamabad Campus, Islamabad 44000, Pakistan
[2]Department of Computer Science, Hazara University Mansehra, Mansehra 21300, Pakistan
[3]Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[4]Department of Computer Science and Engineering, School of Engineering and Technology, Sharda University, Greater Noida 201310, India

Corresponding author: Faiza Habib (faiza.habib@abasynisb.edu.pk)

This Research is funded by Researchers Supporting Project Number (RSPD2024R947), King Saud University, Riyadh, Saudi Arabia.

ABSTRACT Today Internet of Things (IoT) has become a key part of the modern world as it enables web-based IoT devices to collect, transfer, and analyze the data of individuals, companies, and industries. IoT provides numerous services and applications via a massive number of interconnected devices and has become an innovative attack vector for cyber-attacks and threats such as malware attacks that are currently regarded as serious dangers to the security of IoT devices and systems. Such threats are sufficient to infiltrate individual private information that inflicts harm to both the financial standing and reputation in an organization. In literature, researchers have used multiple machine learning and deep learning models to tackle this security threat, however, still accurate classification and detection of metamorphic malware in IoT devices remains a challenge. In this article, we used a deep learning model to accurately detect metamorphic malware in IoT devices. We have employed six models including (VGG16, InceptionV3, CNN, ResNet50, MobileNet, and Efficient NetB0 on Malimg publicly available malware image dataset. The Internet of Things (IoT) would benefit from having a method that could identify metamorphic malware. It isn't possible to rely on detection techniques that are fixed or signature-based. Throughout this research, a straightforward technique for carrying out dynamic analysis to comprehend the behavior of code is suggested. To determine if executable are malicious, it is necessary to first measure the behavior of executable and then utilize this information to make that determination. Additionally, the purpose of this study is to create a classifier that makes utilization of deep learning techniques to analyze complicated behavior reports. The obtained results depict that the proposed model achieves a promising accuracy of 99% and F1-score of 97% employed on the standard Malimg dataset as compared to other existing machine and deep learning models.

INDEX TERMS Internet of Things, IoT security, metamorphic malware detection, deep learning, Malimg dataset, cyber security.

I. INTRODUCTION

Recently, with the advancement in Information Communication Technology (ICT) [1], [2] has enabled the IoT (web of things) technology to make people interact with their homes and workplaces without requiring people-to-computer interaction. The IoT ecosystem [3] has enabled devices (things) such as computing servers, people, digital machines, and networks which are used almost anywhere in the global world to reduce labor and cost via process automation in companies and industries [4]. The scope of IoT applications have enabled the hospitals, businesses,

The associate editor coordinating the review of this manuscript and approving it for publication was Moussa Ayyash.

residences, and organizations [5] to continuously adopt inter-connected IoT devices for increased connectivity and convenience. Conventional computing platforms employ hardware or sensors with a specific function. However, IoT technology need to instantaneously provide entity-level maintenance, logistical, and intelligence data to decision makers empowering them to act with more assurance and speed.

The innovation in ICT technology has enable numerous IoT devices to quickly connect and access the Internet. Due to the openness of the Internet, numerous high-profile attacks and threats are exposing the vulnerability [6] in these inter-connected IoT devices. The security issue has gotten severe as IoT devices have become more widespread [7]. For example, IoT devices are targeted by an average of 5200 attacks per month in the year 2019 [8]. Among these cyber security threats and challenges [32], malware attacks are a danger to these inter-connected devices and network systems over the Internet all around the digital globe.

The remaining article is ordered as follows. Section II overview the existing studies including machine and deep learning models with the different datasets for malware classification and detection in IoT. In Section III, we briefly present the problem statement. Section IV presents our proposed research methodology and solution. Section V analyzes the obtained results and compares them based on performance metrics. Finally, Section VI concludes the proposed work with future directions.

II. BACKGROUND STUDY

Several studies have focused on mitigating and detecting increasing cyber-attacks such as malware attacks on IoT devices using a variety of intrusion detection techniques based on machine learning methods. However, there is limited research work done in visualizing metamorphic malware which presents a significant cyber security challenge to the field of IoT devices and networks. This literature review aims to explore the existing scholarly and practical works on the evolving malicious software (malware) by examining its characteristics, detection, and mitigation approaches. CNN is utilized in the deep learning model known as Inception V3, which is used for pulmonary image classification [12] to produce a better diagnostic model. It is a revised and improved edition of the Inception V1 that was made available to the public for the first time in 2014 under the name GoogleNet [19]. The Inception v3 model that was introduced in 2015 is distinguished from its forerunners by having forty-two layers as well as a smaller margin of inaccuracy as compared to its precedents. In the experiment, different image classifiers (Softmax, SVM, and Logistic) were used on the Japanese Society of Radiological Technology (JSRT) dataset which produces the highest specificity and sensitivity along with better classification performance. The authors in [17] proposed to perform dynamic analysis to detect metamorphic malware instead of using static and/or signature-based methods. The proposed method uses machine learning to understand the behavior of portable executable (PE) or dynamic link libraries (DLLs) via training a classifier that classifies the malware and reports are generated which are further processed to dynamically analyze in order to make a decision whether the PE being malicious or not. However, SVM produces a very low level of malware detection accuracy. The authors in [18] presented an innovative robust cross-architecture MTHAEL model i.e. advanced ensemble learning for new IoT malware detection on various IoT architectures. It utilizes the RNN and CNN for classification which achieves high accuracy and reduces computational time in detection on cross-architecture of IoT network. The proposed model is suitable for a broad variety of IoT applications and forecasting in the event of multiple malware classes However, there is a necessity to research to find associations between various malware and malicious patterns found in different IoT systems. The open-source Malimg dataset is used in the research work [20]. The k-nearest Neighbor (KNN) method is used for malware image classification based on the Euclidean distance. This collection includes gray-scale images of several malware types, but there are just a few image samples and the distribution of images within each class is not uniform. Not all malware types are covered in this research. A novel approach [21] ANN based on a machine learning algorithm is used to detect Mirai and Benign in IoT devices. The Mirai and Benign datasets are used in Matlab2018b implementation and training. The obtained results show significant improvement in achieving better accuracy and false-negative rates in malware detection in IoT systems. The research work in [22] uses TT Analyze tool to dynamically analyze the behavior of unknown Windows executable. The proposed approach tightly examines and monitors the Windows system calls and API calls using an emulated operating system environment which generates a report for system analysts to easily understand the sample by modifying the original code. This approach runs on open-source personal computer (PC) emulators like Qemu2. It does not alter the binaries to prevent being identified as malware rather it focuses on hooks and breakpoints that are present in pertinent APIs and native library code. The proposed tool contributes to achieving high emulating accuracy and helps in to understand quickly the malicious code (unknown malware). The authors in [23] proposed a CNN-based method that minimizes the API sequence (call) length (repetition) which reduces adverse effects on malware classification and detection. The proposed classification method performs convolution operations automatically to get different API call features from the time series sequence (data). Using the information gained, the proposed feature extraction method efficiently and accurately learns the characteristics of API calls and retains its information. The dataset is samples taken from Virus Share, IoTPOT, and TWISC for testing and training the method. The experimental results show the effectiveness of the proposed method with a better classification accuracy of

IoT malware. In [24], authors used malware dynamic analysis to examine and collect data using a system call sequence about the malicious code to monitor its behavior. Deep learning models such as Long-short term memory (LSTM) and nested LSTM are used as classifiers while SVM is used as a benchmark method. In addition, word2vec method is used as word embedding. The malware samples were collected from Virus Share and TheZoo to track the behavior of Cuckoo malware analysis. However, the proposed technique requires a significant sum of malware data to be collected. The problem of a big dataset remains to be solved, even though precision has increased. The authors in [25] used an improved CNN method for malware visualization in the IoT platforms. First, in the proposed method, RGB image feature representation problem between traditional and IoT platforms is solved by paying attention to the developer information and assembly code of the malware. Second, the existing CNN model is improved by combining the spatial pyramid pooling and self-attention mechanism to find out the large size difference in the IoT malware. Experimental results show the effectiveness of the proposed model in cross-platforms for IoT malware variant detection with improved accuracy. A combined DCNN approach in [26] is proposed to detect malware-infected files and pirated software that are serious cyber security threats in the IoT systems. The source code plagiarism technique is applied based on the TensorFlow deep neural network to detect pi-rated source codes. The weighting and tokenization feature technique is used on the dataset samples combined from Malimg and Google Code Jam (GCJ) to filter the noisy data in the plagiarized software (i.e. software piracy). The experimental results show the efficacy of the proposed image visualization method such that software classification performance is enhanced in the IoT network. The authors in [27] investigated the behavioral data of an executable to predict in a short period whether it is malicious and/or Benign or not using the machine learning RNN method. The case study conducted on ransomware samples shows 94% detection accuracy. The authors in [28] proposed a compressed CNN-based residual squeeze Visual Geometry Group (VGG16) which employs 4 convolution layers and 12 fire modules on the MIT Places365-Standard image dataset for image classification, segmentation, and detection. The CNN-based architecture known as VGG16 emerged victorious in the ImageNet contest organized in 2014 by ImageNet Large Scale Visual Recognition Challenge (ILSVR). It is widely considered among the very finest designs for vision models that are currently on the market. The VGG16 prioritizes its three-by-three filter with a stride 1 and employs the similar padding and max pool layer of a two-by-two filter with stride 2 at all times. Across the entirety of the architecture, the max pool and convolution layers are organized similarly and make utilization of SoftMax as an output. The proposed method effectively addresses solving the speed and size while improving aspects like faster speed, faster convergence, smaller model size, better generalization, and enhanced accuracy in image recognition. The authors in [29] proposed a transfer learning based on CNN which uses popular ResNet50 architecture in the pre-trained model to precisely identify the plant diseases. Where ResNet is an abbreviation of Residual Network. It uses the dataset of plant diseases consisting of both disease and healthy plant leaves images. The Resnet50 pertains to a variability that could function using as many as fifty layers in a neural network. ResNet is available in several various forms, all of which have a unique number of layers however with similar fundamental presumptions. The concept ResNet makes it possible to train ultra-deep neural networks. This paradigm asserts that a network could have a large number of layers and nevertheless operate profoundly. The proposed model assures best performance in training accuracy. Efficient-Net is a highly efficient compound scaling model that accomplishes state-of-the-art achievement both in ImageNet and ResNet using basic image categorization transfer learning tasks. Tan and Le initially presented Efficient-Net in 2019, and since then, it has become the greatest successful scaling model [30] which uniformly scales all network dimensions including depth, width, and resolution. Efficient-Net provides a range of models, from B0 to B7 that strike a reasonable balance between the efficiency and precision of their predictions. Scaling heuristics remove the requirement for time-consuming grid-searching of hyper parameters, allowing the B7 models to outclass models at all scales. When compared to different models that accomplish a precision comparable to ImageNet, Efficient-Net is a significantly more compact solution that achieves better accuracy as compared to the state-of-the-art on transfer learning datasets. MobileNet [31] is built with a more condensed architecture than traditional networks. It constructs lightweight DNN by making use of depth-wise distinguishable convolutions. In several applications that are used in the real world, like self-driving cars, robotics, and augmented reality, recognition tasks have to be finished promptly on a platform that is limited in its computational capabilities. MobileNet was introduced by Google and is primarily utilized for mobile application development. Table 2 tabulates the existing research works with intended features regarding the employed malware detection and classification technique/model, and dataset for different malware samples in IoT devices and platforms that are closely related to the proposed research technique in this article.

Table 1 lists the most familiar types of malware according [9], Their variant (a real-world example), and working (danger) which are most likely encountered as:

In the fourth quarter of the year 2016, the Mirai [21] (IoT Botnet) malware attack infected around 80 thousand to 2.5 million Internet-connected devices (computers) via a DDoS attack. Due to the widespread use of search engines, DDoS hackers may easily locate the new IoT devices more quickly over the Internet [10]. Similarly, an economic loss of around 17$ million occurred when the Robbin-Hood

(ransomware) hit the city of Atlanta in the year 2018. To protect and battle with ever-growing cyber security threats and attacks, malware detection has been an important research topic for engineers, academicians, and researchers in recent years [11], [12]. Various machine learning (ML) techniques (supervised, unsupervised, and reinforcement) [13] and artificial intelligence (AI) methods have been employed to enhance security in IoT systems (both host-based and network-based). However, traditional machine learning techniques due to their fixed framework [14] are incompetent to detect and safeguard effectively the IoT devices against the complex cyber threats and vulnerabilities. In contrast, deep learning (DL) techniques and models can quickly detect even the small variants and anomalies in a high-speed network making it highly scalable, cost-efficient, and adaptive without exhausting the resource-constrained IoT devices i.e. a novel breakthrough in IoT cybersecurity. For example, nowadays Recurrent Neural Networks (RNN) are used as dominant deep learning-based malware detection technology [15]. However, RNNs are vulnerable to hostile attacks such as a hacker may carry out a hostile RNN or repetitive API calls. In contrast to existing malware types, metamorphic malware constantly evolves and changes its form (code) over time to deceive traditional detection mechanisms in IoT environments. For example, conventional signature-based detection solutions (e.g., Hash comparison) are useless for detecting metamorphic malware samples and patterns, particularly those that target IoT devices. Measuring the actual behavior (executable activity) and appearance of metamorphic malware to decide whether or not it is malicious, is a challenging task [16] both for practitioners and researchers. Some anti-virus (AV) and malware detection are based on static and dynamic analysis [17] techniques, However, these require a sizable dataset to train the model and are explicit to the type of malware for which they are designed. Convolution neural network (CNN) [18] also provides promising solutions to cope with the security threats posed by metamorphic malware in IoT systems.

Further, CNN is utilized in the deep learning model known as Inception V3, which is used for pulmonary image classification [12] to produce a better diagnostic model. It is a revised and improved edition of the Inception V1 that was made available to the public for the first time in 2014 under the name GoogleNet [19]. The Inception v3 model that was introduced in 2015 is distinguished from its forerunners by having forty-two layers as well as a smaller margin of inaccuracy as compared to its precedents. In the experiment, different image classifiers (Softmax, SVM, and Logistic) were used on the Japanese Society of Radiological Technology (JSRT) dataset which produces the highest specificity and sensitivity along with better classification performance. The authors in [17] proposed to perform dynamic analysis to detect metamorphic malware instead of using static and/or signature-based methods. The proposed method uses machine learning to understand the behavior of portable executables (PE) or dynamic link libraries (DLLs) via training a classifier that classifies the malware and reports are generated which are further processed to dynamically analyze in order to make decision whether the PE being malicious or not. However, SVM produces a very low level of malware detection accuracy. The authors in [18] presented an innovative robust cross-architecture MTHAEL model i.e. advanced ensemble learning for new IoT malware detection on various IoT architectures. It utilizes the RNN and CNN for classification which achieves high accuracy and reduces computational time in detection on cross-architecture of IoT network. The proposed model is suitable for a broad variety of IoT applications and forecasting in the event of multiple malware classes However, there is a necessity to research to find associations between various malware and malicious patterns found in different IoT systems. The open-source Malimg dataset is used in the research work [20]. K-Nearest Neigh-bor (KNN) method is used for malware image classification based on the Euclidean distance. This collection includes grayscale images of several malware types, but there are just a few image samples, and the distribution of images within each class is not uniform. Not all malware types are covered in this research. A novel approach [21] ANN based on a machine learning algorithm is used to detect Mirai and Benign in IoT devices. The Mirai and Benign datasets are used in Matlab2018b implementation and training. The obtained results show significant improvement in achieving better accuracies and false-negative rates in malware detection in IoT systems. The research work in [22] uses the TTAnalyze tool to dynamically analyze the

TABLE 1. Common examples of malware types, variants and its working.

Type(s)	Variant	Working
Adware	Fireball	Tracks user's activity on Internet and display unwanted advertisements
Bots/Botnet	Echobot	Propagates a broad number of remote-controlled attacks (e.g., Botnet)
Fileless Malware	Astaroth	Edits the native files of an operating system (e.g., Power Shell)
Keyloggers	Olympic Vision	Monitors keystrokes to steal user's passwords (e.g., Business and email compromises attack)
Mobile Malware	Triada	Infects mobile phones (Android devices) via installing spam applications
Ransomware	RYUK	Disables target (victim's) access to its data until payment (ransom) is paid (e.g., RobbinHood)
Rootkits	Zacinlo	Give attackers to remotely control a victim's computer (shared device) with full privileges
Spyware	DarkHotel	Collects victim's activity data (e.g., passwords) without their consent via browser, critical apps in mobile phones
Trojan	Emotet	Personate itself as desirable software (code) and take control for malicious activities
Virus	SQL	Slammer It infects computer to harm their data and software
Worms	Stuxnet	Spreads through flash drives or software into a network to launch DDoS or steal confidential data
Wiper Malware	Whisper	Gate Ensures that user data is erased that cannot be recovered

behavior of an unknown Windows executable. The proposed approach tightly examines and monitors the Windows system calls and API calls using an emulated operating system environment which generates a report for system analysts to easily understand the sample by modifying the original code. This approach runs on open-source personal computer (PC) emulators like Qemu2. It does not alter the binaries to prevent being identified as malware rather it focuses on hooks and breakpoints that are present in pertinent APIs and native library code. The proposed tool contributes to achieving high emulating accuracy and helps in to understand quickly the malicious code (unknown malware). The authors in [23] proposed a CNN-based method that minimizes the API sequence (call) length (repetition) which reduces adverse effects on malware classification and detection. The proposed classification method performs convolution operations automatically to get different API call features from the time series sequence (data). Using the information gained, the proposed feature extraction method efficiently and accurately learns the characteristics of API calls and retains its information. The dataset is samples taken from Virus share, IoTPOT, and TWISC for testing and training the method. The experimental results show the effectiveness of the proposed method with a better classification accuracy of IoT malware. In [24], authors used malware dynamic analysis to examine and collect data using a system call sequence about the malicious code to monitor its behavior. Deep learning models such as Long-short term memory (LSTM) and nested LSTM are used as a classifier while SVM is used as a benchmark method. In addition, the word2vec method is used as word embedding. The malware samples were collected from Virus Share and The Zoo to track the behavior of Cuckoo malware analysis. However, the proposed technique requires a significant sum of malware data to be collected. The problem of a big dataset remains to be solved, even though precision has increased. The authors in [25] used an improved CNN method for malware visualization in the IoT platforms. First, in the proposed method, RGB image feature representation problem between traditional and IoT platforms is solved by paying attention to the developer information and assembly code of the malware. Second, the existing CNN model is improved by combining the spatial pyramid pooling and self-attention mechanism to find out the large size difference in the IoT malware. Experimental results show the effectiveness of the proposed model in cross-platforms for IoT malware variant detection with improved accuracy. A combined DCNN approach in [26] is proposed to detect malware-infected files and pirated software that are serious cybersecurity threats in IoT systems. The source code plagiarism technique is applied based on the TensorFlow deep neural network to detect pi-rated source codes. The weighting and tokenization feature technique is used on the dataset samples combined from Malimg and Google Code Jam (GCJ) to filter the noisy data in the plagiarized software (i.e. software piracy). The experimental results show the efficacy of the proposed image visualization method such that software classification performance is enhanced in the IoT network. The authors in [27] investigated the behavioral data of executable to predict in a short period whether it is malicious and/or Benign or not using the machine learning RNN method. The case study conducted on ransomware samples shows 94% detection accuracy. The authors in [28] proposed a compressed CNN-based residual squeeze Visual Geometry Group (VGG16) which employs 4 convolution layers and 12 fire modules on the MIT Places365-Standard image dataset for image classification, segmentation, and detection. The CNN-based architecture known as VGG16 emerged victorious in the ImageNet contest organized in 2014 by ImageNet Large Scale Visual Recognition Challenge (ILSVR). It is widely considered among the very finest designs for vision models that are currently on the market. The VGG16 prioritizes its three-by-three filter with a stride 1 and employs the similar padding and max pool layer of a two-by-two filter with stride 2 at all times. Across the entirety of the architecture, the max pool and convolution layers are organized in a similar manner and make utilization of SoftMax as an output. The proposed method effectively addresses solving the speed and size while improving aspects like faster speed, faster convergence, smaller model size, better generalization, and enhanced accuracy in image recognition. The authors in [29] proposed a transfer learning based on CNN which uses popular ResNet50 architecture in the pre-trained model to precisely identify the plant diseases. Where ResNet is an abbreviation of Residual Network. It uses the dataset of plant diseases consisting of both disease and healthy plant leaves images. The Resnet50 pertains to a variability that could function using as many as fifty layers in a neural network. ResNet is available in a number of various forms, all of which have a unique number of layers however with similar fundamental presumptions. The concept ResNet makes it possible to train ultra-deep neural networks. This paradigm asserts that a network could have a large number of layers and nevertheless operate profoundly. The proposed model assures bet performance in training accuracy. EfficientNet is a highly efficient compound scaling model that accomplishes state-of-the-art achievement both in ImageNet and ResNet using basic image categorization transfer learning tasks. Tan and Le initially presented EfficientNet in 2019, and since then, it has become the greatest successful scaling model [30] which uniformly scales all network dimensions including depth, width, and resolution. EfficientNet provides a range of models, from B0 to B7 that strike a reasonable balance among the efficiency and precision of their predictions. Scaling heuristics remove the requirement for time-consuming grid-searching of hyperparameters, allowing the B7 models to outclass models at all scales. When compared to different models that accomplish a precision comparable to ImageNet, EfficientNet is a significantly more compact solution that achieves better accuracy as compared to the state-of-the-art on transfer learning datasets. MobileNet [31] is built with a more condensed architecture than traditional networks.

TABLE 2. Comparison and summarization of existing research works of malware detection in IoT devices.

Ref	Method	Dataset	Feature(s)
[12]/19	Inception V3	JSRT	High sensitivity and better performance
[17]/15	SVM	Dynamic	Report Metamorphic malware Classification and detection
[18]/20	MTHAEL	Cross-architecture IoT	High accuracies and low computational overheads
[20]/11	KNN	Malimg	Classification accuracy
[21]/21	ANN	Mirai and Benign	Achieves better accuracy and false-negative rate
[22]/06	TTAnalyze tool	Windows and Native API calls	Quick understanding and reducing vulnerability window
[23]/20	CNN	IoTPOT, TWISC, and VirusShare	Faster classification and higher accuracy
[24]/20	Word2vec	VirusShare and TheZoo	Lightweight and accuracy is enhanced
[25]/21	CNN	Traditional and IoT	Improved accuracy
[26]/19	DCNN	GCJ and Malimg	Enhanced classification performance
[27]/18	RNN	Virus Total and Virus Share	Higher accuracy
[28]/18	VGG16	Places365-Standard Smaller	model size, faster recognition, and better accuracy
[29]/19	Resnet50	Plant disease	Best training accuracy
[30]/19	Efficient Net-B7	Image-Net and ResNet	Improves accuracy
[31]/23	MobileNet	ImageNet	Improve performance
our/23	DCNN	Malimg	Higher Accuracy and improve performance

It constructs lightweight DNN by making use of depth-wise distinguishable convolutions. In several applications that are used in the real world, like self-driving cars, robotics, and augmented reality, recognition tasks have to be finished promptly on a platform that is limited in its computational capabilities. MobileNet was introduced by Google and is primarily utilized for mobile application development.

Table2 tabulates the existing research works with intended features regarding the employed malware detection and classification technique/model, the dataset for different malware samples in IoT devices and platforms that are closely related to the proposed research technique in this article

A. INCEPTIONV3

INCEPTIONV3 is a convolutional neural network (CNN)–based deep learning model designed for image classification. It represents a significant step forward from its predecessor, Inception V1, also known as GoogLeNet, which debuted in 2014. Released in 2015 [19], Inception V3 features 42 layers and achieves a smaller margin of error compared to previous versions. InceptionV3 is known for its ability to efficiently capture features at multiple scales, which can be vital for classifying various patterns and structures within the malware images. Its deep architecture permits it to learn complex representations, which can be useful in distinguishing between different types of malware.

B. ResNet50

ResNet, also known as Residual Network, encompasses several forms, with ResNet50 denoting a variant capable of accommodating up to 50 layers of neural networks. The fundamental principle of ResNet allows the training of extremely deep neural networks. He revolutionized the field by demonstrating that networks with hundreds or even thousands of layers could maintain high levels of performance. ResNet50's architecture, with its residual connections, allows for the training of very deep networks without suffering from the vanishing gradient problem. Malware images can vary significantly in complexity and structure, and ResNet50's ability to capture fine-grained details across many layers can be advantageous for classification tasks involving such diverse visual content.

C. EFFicientNet-B0

EfficientNet, introduced by Tan and Le in 2019, is recognized for its state-of-the-art performance in image transfer learning and classification tasks. It includes a range of models from B0 to B7, striking a balance between efficiency and precision. The base model, B0, prioritizes efficiency and outperforms models at different scales using scaling heuristics, thus rationalizing the need for extensive parameter tuning. EfficientNetB0 strikes a balance between model size and accuracy, making it suitable for resource-constrained environments. Malware classification tasks may require deploying models on devices with limited computational resources. EfficientNet-B0's smaller size compared to other models allows for efficient inference while still achieving competitive performance.

D. MOBILE-NET

Mobile-Nets uses a compact architecture built with depth-wise separable convolutions, making it easy to build lightweight deep neural networks (DNNs). This design addresses the demand for rapid recognition tasks on limited computing platforms in real-world applications such as autonomous vehicles, augmented reality, and robotics. Developed by Google, Mobile-Net is designed primarily for mobile applications, emphasizing efficiency and performance in resource-constrained environments. Mobile-Net compact architecture, designed specifically for mobile and embedded devices, can be advantageous for real-time malware classification applications, especially in scenarios where processing power and memory are limited. Its lightweight design and efficient depth-wise separable convolutions make it well-suited for deployment on mobile platforms.

III. CHALLENGES OF THIS RESEARCH

The rise of IoT devices has revolutionized various sectors e.g., industries by providing increased connectivity and

convenience between networks, systems, and users. However, this rapid growth has also introduced new security challenges, with metamorphic malware emerging as a significant threat to the integrity and security of IoT devices. Unlike traditional malware, metamorphic malware can dynamically change its code and structure, making it highly elusive and challenging to detect and mitigate in the systems. Existing studies in the literature have proposed different techniques to classify and detect security threats in IoT networks. However, there is a need for effective countermeasures against metamorphic malware in IoT devices, emphasizing the potential consequences of these attacks, the limitations of current security measures, and the urgency to develop robust models to examine and monitor whether a software code and/or behavioral data is malicious or not with a higher accuracy such that to safeguard the IoT devices and systems. Hence, how to make secure IoT devices from evolving metamorphic malware with improved accuracy and performance? This is the research question that needs to be solved and the scope of this study in this article.

IV. RESEARCH METHODOLOGY

This section describes the proposed methodology.

A. ANALYSIS OF DATASET

In this work, the open-source Malimg (Malware Image) dataset [20] is used as the primary information source in our proposed research methodology. The dataset consists of 8 different classes with 25 different families having 9,342 unique samples of malware as listed in Table 3. Malimg dataset is a valuable resource effectively used in the classification and detection of metamorphic malware in IoT devices. It empowers researchers, developers, and security professionals to develop more adaptive and robust defense mechanisms for IoT platforms to cope with ever-evolving cyber threats.

The Malimg dataset includes a comprehensive collection of gray-scale images representing various types of malware samples, representing a significant number of images. This abundance allows for nuanced exploration of malware classification tasks through visual perspectives, leveraging computer vision and image processing techniques. With its variety of malware types and classification challenges, the dataset accurately reflects real-world scenarios, reflecting the evolving cyber threat landscape. Its large volume facilitates the development and evaluation of robust machine learning and deep learning models for automated malware detection and classification, thereby addressing the imperative need for effective cyber security measures. Additionally, by serving as a benchmark for comparing classification algorithms and evaluating feature extraction methods, the dataset accelerates advances in malware analysis and detection, strengthening defenses against cyber threats.

B. DATA PRE-PROCESSING

Pre-processing is the initial step in the deep learning pipeline. The sole aim of this step is to translate raw input into a comprehensible to model. This phase ensures data suitability and enhancement of productivity. This phase involves the manipulation of inputs like resizing images to match the dimensions of input layers. The pre-processing step can optimize certain desirable traits and eliminate artifacts that may compromise network performance.In this work, we have employed the pre-processing step before inputting samples into training networks. Initially, the sample images are transformed into 8-bit vector images through a method [17] which converts them into gray-scale matrices. These 8-bit values range from 0(black) to 255(white).

TABLE 3. Malimg dataset information.

No.	Class	Family	Sample
01	Worm	Allaple.L	1591
02	Worm	Allaple.A	2949
03	Worm	Yuner.A	800
04	PWS	Lolyda.AA 1	213
05	PWS	Lolyda.AA 2	184
06	PWS	Lolyda.AA 3	123
07	Trojan	C2Lop.P	146
08	Trojan	C2Lop.gen!G	200
09	Dialer	Instantaccess	431
10	Trojan Downloader	Swizzor.gen!I	132
11	Trojan Downloader	Swizzor.gen!E	128
12	Worm	VB.AT	408
13	Rogue	Fakerean	381
14	Trojan	Alueron.gen!J	198
15	Trojan	Malex.gen!J	136
16	PWS	Lolyda.AT	159
17	Dialer	Adialer.C	125
18	Trojan Downloader	Wintrim.BX	97
19	Dialer	Dialplatform.B	177
20	Trojan Downloader	Dontovo.A	162
21	Trojan Downloader	Obfuscator.AD	142
22	Backdoor	Agent.FYI	116
23	Worm:AutoIT	Autorun.K	106
24	Backdoor	Rbot!gen	158
25	Trojan	Skintrim.N	80

C. FEATURE ENGINEERING

The recognition of malware families can be performed by leveraging the most prominent feature patterns existing in the malware images. Images from the same class or family tend to exhibit similar patterns, which is helpful for the CNN model to extract features automatically. In our proposed method, we have extracted different features using CNN architecture. The CNN layers are exploited to learn the different representations of malware to improve the classifier's performance. During the training phase, five layers of the model were trained from scratch on image representation of all the different malware classes. During this process, the best model state is saved through which log loss is minimized

D. MODEL TRAINING STEPS

The most important steps in our model training procedure are illustrated in Figure1. Every individual step plays an essential part in achieving the best possible performance outcome for every algorithm. We have trained different

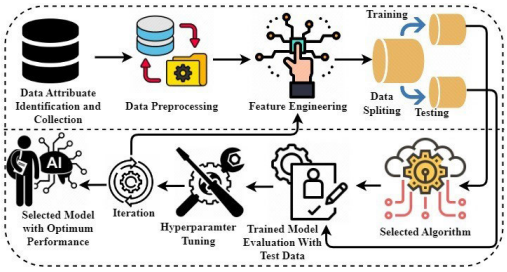


FIGURE 1. Model training steps flow diagram.

CNN models for classifying malware images on the Malimg dataset. We have trained CNN, VGG16, InceptionV3, ResNet50, Mobile-Net, and EfficientNet-B0. Each of these models has different architectures and characteristics. CNN is the foundational model in computer vision having basic architecture designed for image classification, that consists of multiple convolutional layers followed by a pooling layer and a fully connected layer.

We have trained VGG16 with 16 layers having many small convolutional filters. This model is very well known for its simplicity and effectiveness for different computer vision tasks. InceptionV3 also known as GoogLeNet is popular for its inception modules which is based on multiple filter sizes in parallel to extract features and various scales. ResNet50 contains 50 layers and it is known for mitigating the vanishing gradient problem. It has introduced residual connections that are suitable for the training of deep networks effectively. Mobile net is designed to be efficient in mobiles and embedded devices. It can separate convolutions depth-wise which leads to reducing the number of computations and parameters while maintaining good accuracy. EfficientNet B0 is a prominent member of the EfficientNet family, it is a scaled-down version that aims to balance model size and accuracy. EfficientNetB0 exploits the compound scaling technique for the optimization of architecture based on the resolution, with dth and depth.

The architecture of CNN primarily consists of three basic components (shown in fig2) i.e. the convolutional layer, the pooling layer and the fully connected layer. In this research, we have employed different CNN models with the sequence of convolutional layers filters of varied sizes and maintained a consistent stride of extracted feature maps. The output of each convolutional layer is channeled through the ReLU activation function and subjected to the max-pooling layer with two strides.At last, these layers are succeeded by a fully connected layer, and a sigmoid activation function is used to map these features into a single vector. Our techniques diverge from the conventional CNNs classification task in that we have added a custom layer to compute the Manhattan distance between different feature vectors. This process results in output which signifies the similarity score of a pair of malware images.

We have trained VGG16 with 16 layers having many small convolutional filters. This model is very well known for its simplicity and effectiveness for different computer vision

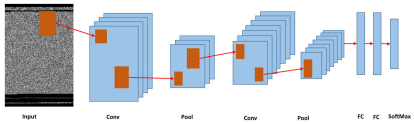


FIGURE 2. Convolutional stages.

tasks. InceptionV3 also known as GoogLeNet is popular for its inception modules which are based on multiple filter sizes in parallel to extract features and various scales. ResNet50 contains 50 layers and it is known for mitigating the vanishing gradient problem. It has introduced residual connections that are suitable for the training of deep networks effectively. Mobile net is designed to be efficient in mobiles and embedded devices. It can separate convolutions depthwise which leads to reducing the number of computations and parameters while maintaining good accuracy. EfficientNet B0 is a prominent member of EfficientNet family, it is a scaled-down version that aims to balance model size and accuracy. EfficientNetB0 exploits compound scaling technique for the optimization of architecture based on the resolution, withdth, and depth.

The architecture of CNN primarily consists of three basic components (shown in fig2) i.e. the convolutional layer, the pooling layer and the fully connected layer. In this research, we have employed different CNN models with the sequence of convolutional layers filters of varied sizes and maintained a consistent stride of extracted feature maps. The output of each convolutional layer is channeled through the ReLU activation function and subjected to the max-pooling layer with two strides. At last, these layers are succeeded by a fully connected layer and a sigmoid activation function is used map these features into a single vector. Our techniques diverge from the conventional CNNs classification task in that we have added custom layer to compute the Manhattan distance between different feature vectors. This process results in output which signifies the similarity score of a pair of malware images.

E. TRAINING AND TESTING DATA SPLIT

The training process for these models is conducted by using small batches. For effective training, the model's image pairs are randomly chosen, while the unbalanced pairs of images are avoided. The dataset is split into 80% for training while the remaining 20% are used for testing purposes. For addressing any ethical consideration, especially data privacy, we ensure that the Malimg dataset may not contain personally identifiable information and that data collection from IoT devices is performed securely.

F. PERFORMANCE EVALUATION MEASURES

The following performance evaluation matrices are used to evaluate and compare different classification models all of which emerge with a distinct individual set of advantages and disadvantages, including F1-score, precision, recall, false

positive rate (FPR), and accuracy can be described as:
4.6.1 F1-Score

The F1-score, additionally known as the "F score," is a method that can be used to evaluate the accuracy of a model based on a dataset. It is utilized in the process of estimation and evaluation of binary classification. The F1-score represents the arithmetic mean of both recall and precision which ranges from 0 to 1 where o indicates the worst performance and 1 represents perfect precision and recall. It is employed to judge the accuracy of statistical measurements. Subsequently, F1-score can vary from zero to one that can be formulated in Eq. (1) as:

$$\frac{2 * (Precision - Recall)}{Precision + Recall} \tag{1}$$

G. PRECISION, RECALL AND FALSE POSITIVE RATE

To assess the classification models using unbalanced datasets, two metrics that can be helpful are precision and recall. Where precision that is also known as positive predictive values can be computed as the true positives (TP) prediction to the total predicted positives values (i.e. TP+ FP (false positives)) in Eq. (2) as:

$$\frac{TP}{(TP + FP)} \tag{2}$$

Similarly, recall also known as sensitivity can be calculated as a true positive (TP) to the total actual positives (i.e. TP+ FN (false negative)).

H. ACCURACY

Accuracy is the most commonly used rational performance metric to evaluate the performance of a classification model which can be measured as the proportion of correctly predicted outcomes to the total evaluation (number of instances) in a dataset. Accuracy can be characterized in Eq. (3) as:

$$\frac{(TP + TN)}{(FP + FN + TP + TN)} \tag{3}$$

where TN represents the true negatives which are predicted correctly as negative. All these relevant metrics are considered to assess and understand the overall effectiveness and performance of model behavior in a specific IoT scenario.

I. LABEL ENCODER

Label encoding refers to the procedure of transforming labels from their human-readable form into a numerical format that is readable by machines. It's a technique of encoding that is frequently utilized for categorical values. After that, algorithms that use deep learning could make more informed decisions regarding the way these labels must be applied. When it comes to supervised learning, this pre-processing step for the structured dataset is extremely important. In this method, an individual integer is assigned to every label, and the assignment is determined by the label's position in the alphabet.

```
{'Fakerean': 381, 'Instantaccess': 431, 'Allaple.L': 1591, 'Allaple.A': 294
9, 'Agent.FYI': 116, 'Dialplatform.B': 177, 'Dontovo.A': 162, 'Lolyda.AA2':
184, 'Yuner.A': 800, 'Lolyda.AT': 159, 'Obfuscator.AD': 142, 'Alueron.gen!
J': 198, 'Swizzor.gen!E': 128, 'Rbot!gen': 158, 'Swizzor.gen!I': 132, 'Adia
ler.C': 122, 'Wintrim.BX': 97, 'VB.AT': 408, 'Malex.gen!J': 136, 'Lolyda.AA
1': 213, 'C2LOP.P': 146, 'C2LOP.gen!g': 200, 'Lolyda.AA3': 123, 'Autorun.
K': 106, 'Skintrim.N': 80}
```

FIGURE 3. Class name before labeling.

```
{10: 381, 11: 431, 3: 1591, 2: 2949, 1: 116, 8: 177, 9: 162, 13: 184, 24: 8
00, 15: 159, 17: 142, 4: 198, 20: 128, 18: 158, 21: 132, 0: 122, 23: 97, 2
2: 408, 16: 136, 12: 213, 6: 146, 7: 200, 14: 123, 5: 106, 19: 80}
```

FIGURE 4. Class name after labeling.

Figure 3 demonstrates the classes before the application of the label encoder, while Figure 4 demonstrates the classes following the application of the label encoder.

J. EXPERIMENTAL SETUP

To achieve the best classification results we have exploited the best available resources to achieve our goal. Therefore, we used Nvidia GTX 960 GPU to speed up the learning process of the models along with 16 GB RAM and Intel Core i7 - 6700HQ 2.6 Processor. The software packages include Jupyter Notebook and Python 3.8 while Tensorflow, OpenCV and SKlearn, Numpy, and Matplotlib libraries were utilized. Experimental parameters i.e. learning rate:0.01, Optimizer: Adam were used.

V. ANALYSIS OF RESULTS

In this section, we analyze and compare the results based on the performance metrics to evaluate the performance of our proposed model with existing research methods (models) for metamorphic malware detection in IoT devices. Table 4 tabulates the obtained results of different models using the performance metrics F1-score, precision, recall, and model accuracy (Acc:). For example, our proposed model has achieved an F1-score of 97% which is better than the other existing models that is, MobileNet (96%), CNN (96%), Resnet50 (94%), and InceptionV3 (89%) respectively. Whereas, VGG16 has the lowest F1-score of 59% in comparison to all these models. Overall, our proposed model outperforms the existing models in the classification and detection of metamorphic malware in IoT devices. The figure shows the accuracy plot of the proposed model EfficientNetB0 and the other models VGG16, MobileNet, ResNet50, and InceptionV3 on the malware image dataset Malimg. We have also performed a comparative analysis of our proposed model with the state-of-the-art works present in the literature. We have observed that the results of EfficientNetB0 have outperformed the rest of the models we have trained and tested.

The ResNet50 and MobileNet have also achieved comparable accuracy with the proposed model as shown in table 4. The VGG model has performed poorly up to 15 epochs after that we can see a little bit of improvement in the performance. The reason behind the poor performance of VGG16 is that it

TABLE 4. Summary and comparison of performance metrics values with existing models.

Ref:	Model	F1	Precision	Recall	Model Acc:
[12]	InceptionV3	89%	90%	89%	96.22%
[25]	CNN	96%	96%	96%	98.50%
[28]	VGG16	59%	62%	63%	91.97%
[29]	Resnet50	94%	94%	95%	98.93%
[31]	MobileNet	96%	97%	97%	98.61%
oue	EfficientNetB0	97%	98%	97%	99.14%

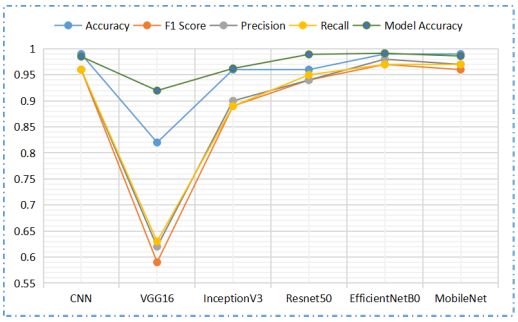


FIGURE 5. Performance metrics graph.

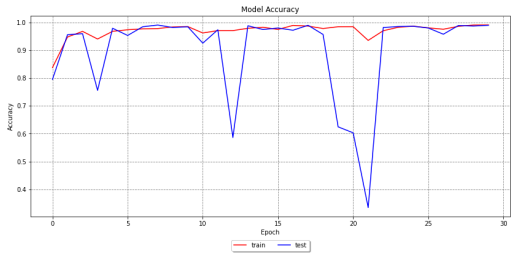


FIGURE 6. ResNet50 model accuracy.

uses basic pooling and fully connected layers and was unable to extract the finest features from the malware image. CNN and InceptionV3 have performed better than the VGG16. We believe that the models that performed better have deep layers and are used especially for image classification even if it contains complex patterns.

Figure 5 illustrates the performance results of the proposed and existing models tabulated in Table 3. It could be noticed that the proposed model produced encouraging performance results.

In accordance with Table 4, the images that make up for every category are not identical. One possible explanation is that the dispersion of images isn't really uniform across the entire class.

As can be seen in Table 4 and Figure 6, the F1 Score, Accuracy, Recall, Precision, and Model Accuracy of these 3 models are extremely comparable to one another.

In the figure, we can see that the proposed EfficientNetB0 has a negligible loss. EfficientNetB0 has performed better than the rest of the models shown in Figure 5. In Figure 6 and Figure 7 the accurcay of ResNet50 and EfficientNetB0 is shown respectively. These two models have the closest accuracy. The accuracy of ResNet50 becomes consistent after

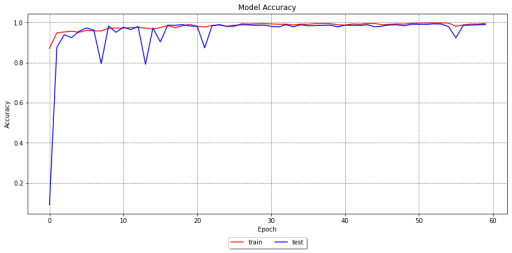


FIGURE 7. EfficientNetB0 model accuracy.

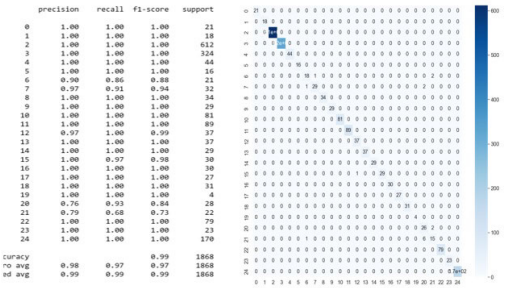


FIGURE 8. Confusion matrix of efficientNetB0.

30 epochs while the accuracy of EfficientNetB0 is 99% after 20 epochs which indicates that Efficient Net takes fewer epochs to reach the maximum accuracy from the rest of the models. Although MobileNet and inceptionV3 also performed well as compared to CNN and VGG16. Except VGG16 all the other models achieved more than 97% accuracy. The efficientNetB0 has a loss between 0 to 0.1 after 20epochs. The other models have obtained a higher loss rate as compared to efficientNetB0.

Once it involves assessing model performance in the manner of a matrix, the confusion matrix technique is among the most effective methods that can be used. It's a two-dimensional A to B matrix, with A representing the no. of precise classes and B representing the no. of predictive classes, appropriately. The confusion matrix compares the actual target classes to the classes that the model anticipated they would be. This provides a thorough perspective as well as information on the efficiency in addition to the errors made by the classification model. The explanation provided by the relevant class is used to ascertain the metrics. In the process of evaluating algorithms for the detection of malware, conveying every measure with regard to every family uncovers the strengths and limitations of the model. Figure 8 represents the confusion matrix diagram of the model proposed for malware classification figure clearly indicates that the performance of efficientNetB0 is better than the rest of the models..

A. COMPARISON OF RESULTS

Throughout this portion, we try comparing the results that we acquired with the findings of earlier research that was carried out by a variety of authors. We compared our EfficientNetB0 results with those from the four research studies [21], [33], [34], [35], [36] as indicated in Table 5, and the graphical

TABLE 5. Summary and comparison of performance metrics values with existing models.

Ref:	Model *Acc:	F1	Precision	Recall	
[21]	ANN	93%	95%	92%	99%
[33]	CNN	98%	98%	99%	96%
[34]	LSTM	98%	97%	97%	97%
[35]	Attention,SPP,RGB	98%	98%	98%	98%
[36]	DNN	98%	97%	97%	97%
Our	EfficientNetB0	99%	97%	98%	97%

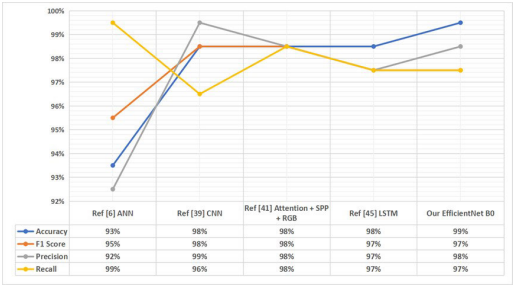


FIGURE 9. Comparison work graphical representation.

representation of the comparison work in Figure 7. Table 5 makes it quite evident that our suggested model outperforms other models.

Comparison research has shown that colored images [30] can have a substantial impact on how well DNN outcomes function. During the training phase, the RGB virus image increased the computational load on the machine.

Since most research methods convert malware binary section information into gray-scale images, this work has advanced further from earlier related research works. In this study, we contend that color images can be more useful for classifying and identifying variants. The primary premise is that color photos contain natural color images and have sharper characteristics than gray-scale, which is expected to improve the efficiency and precision of detection and classification. The binary representation of the original virus is converted by our model into all of color and grayscale images. Throughout this section, we evaluate how well our method performs on color and black-and-white photographs side by side. We chose two Malimg datasets, one in color and the other in gray scale, both of which had identical sample distribution all over malware families. We discovered that gray-scale samples had a higher overall classification accuracy than color samples. In this study, we evaluated the DCNN performance to that of pre-trained networks like VGG16, ResNet50, and InceptionV3. The multi-class malware family classification challenge has previously been resolved using these cutting-edge designs. A total classification precision of 97.12 percent was achieved for VGG16 in this experiment, which is significantly lower as compared to the accuracy of 98.82 percent achieved by DCNN, 98.61 percent achieved by ResNet50, and 98.65% achieved by InceptionV3, that is only slightly lower than the DCNN accuracy of 98.82%, the other performance metrics. We contrast the performance of the DCNN with that of a prior work on malware classification that made use of the machine and deep learning-based image-based malware classification algorithms. Prior to applying machine learning or deep learning classifiers (such as Softmax, KNN, or SVM) for multi-class classification, these techniques retrieved characteristics from the malware images. In this work, we have proposed an Efficient-NetB0 framework for the classification of malware images. We have evaluated the performance of the model as well the parameters used by the model. In addition, we have performed an extensive evaluation by using pre-trained models available in the prior. We have noticed during the comparison of the results that malware using byte-level representation achieved better results as compared to the image representation with byte-encode-based image width selection.

In last, this research work can be practically applied in various industrial environments such as manufacturing plants, Smart Grid infrastructure, transportation, logistics, smart buildings, and healthcare, etc., where IoT devices are confronted with malware and cyber threats. By deploying the proposed intrusion detection system using DNNs based on the Malimg dataset, IoT devices in industrial deployments will be able to proactively mitigate and detect various security risks, enhancing the integrity, reliability, and resilience of the interconnected IoT networks.

VI. CONCLUSION

The advancement in ICT technologies has increased the use of IoT devices in human lives over the public network (Internet). This easy access over the public shared network to these IoT devices has made vulnerable the digital landscape to cyber threats and attacks making the IoT communication system challenging and complex to secure and protect it. In this article, we employed a deep learning technique named, DCNN to classify and detect metamorphic malware (viruses) patterns to protect IoT devices from these malware that could evolve rapidly in less time in the dynamic IoT platforms. We used six different deep learning models including CNN, VGG16, InceptionV3, Resnet50, Mo-bileNet, and EfficientNetB0 to experiment on the Malimg dataset for comprehensive evaluation of metamorphic malware in IoT devices. The experimental results of these models are computed and compared using the performance metrics. The obtained results show that EfficientNetB0 model has produced promising results as compared to the other machine learning and deep learning models employed in the previous research works. In the context of metamorphic malware detection, EfficientNetB0 contributes significantly in the detection and complex feature extraction process which enhances the detection accuracy and improves the performance to tackle these evolving security threats and attacks causing damage. For example, we found in the results that EfficientNetB0 has achieved a very encouraging detection accuracy of 99% (around), F1-score 97%, precision 98%, and recall 97% respectively when applied to the test dataset.

VII. FUTURE WORK

Our future studies will extend the current work by developing a hybrid technique integrating machine and deep learning methods that will be capable of recognizing and categorizing metamorphic malware on a wide range of complicated dataset in different resource constrained IoT-enabled environment including, smart grid, smart health systems, and smart city. Future malware detection work using Malimg dataset could focus on improving feature representation, exploring ensemble learning methods, improving robustness adversaries, leveraging transfer learning, and solving scalability and efficiency issues. Limitations of using models for malware classification include difficulties in generalization to unseen malware samples, class imbalance issues, vulnerabilities to adversarial attacks, and difficulties in interpretability and explainability.

CONFLICT OF INTEREST

The authors declares no conflict of interest.

REFERENCES

[1] S. N. Swamy and S. R. Kota, "An empirical study on system level aspects of Internet of Things (IoT)," *IEEE Access*, vol. 8, pp. 188082–188134, 2020.

[2] M. Brahma, M. A. Rejula, B. Srinivasan, S. N. Kumar, W. A. Banu, K. Malarvizhi, S. S. Priya, and A. Kumar, "Learning impact of recent ICT advances based on virtual reality IoT sensors in a metaverse environment," *Meas., Sensors*, vol. 27, Jun. 2023, Art. no. 100754.

[3] M. R. Prathyusha and B. Bhowmik, "IoT-enabled smart applications and challenges," in *Proc. 8th Int. Conf. Commun. Electron. Syst. (ICCES)*, Jun. 2023, pp. 354–360, doi: 10.1109/ICCES57224.2023.10192597.

[4] D. Peraković, M. Periša, and P. Zorić, "Challenges and issues of ICT in industry 4.0," in *Advances in Design, Simulation and Manufacturing II, Proceedings of the 2nd International Conference on Design, Simulation, Manufacturing: The Innovation Exchange*, pp. 259–269.

[5] A. Khan, A. I. Umar, S. H. Shirazi, W. Ishaq, M. Shah, M. Assam, and A. Mohamed, "QoS-aware cost minimization strategy for AMI applications in smart grid using cloud computing," *Sensors*, vol. 22, no. 13, p. 4969, Jun. 2022, doi: 10.3390/s22134969.

[6] N. Mangala and K. R. Venugopal, "Short paper: Current challenges in IoT cloud smart applications," in *Proc. IEEE Int. Conf. Cloud Comput. Emerg. Markets (CCEM)*, Oct. 2021, pp. 36–40, doi: 10.1109/CCEM53267.2021.00016.

[7] B. Vignau, R. Khoury, and S. Hallé, "10 years of IoT malware: A feature-based taxonomy," in *Proc. IEEE 19th Int. Conf. Softw. Qual., Rel. Secur. Companion (QRS-C)*, Jul. 2019, pp. 458–465, doi: 10.1109/QRS-C.2019.00088.

[8] J. A. Faysal, S. T. Mostafa, J. S. Tamanna, K. M. Mumenin, M. M. Arifin, M. A. Awal, A. Shome, and S. S. Mostafa, "XGB-RF: A hybrid machine learning approach for IoT intrusion detection," *Telecom*, vol. 3, no. 1, pp. 52–69, Jan. 2022, doi: 10.3390/telecom3010003.

[9] K. Baker. *The 12 Most Common Types of Malware*. Accessed: Aug. 15, 2023. [Online]. Available: https://www.crowdstrike.com/cybersecurity-101/malware/types-of-malware/

[10] N. Vlajic and D. Zhou, "IoT as a land of opportunity for DDoS hackers," *Computer*, vol. 51, no. 7, pp. 26–34, Jul. 2018, doi: 10.1109/MC.2018.3011046.

[11] Q. Zhang and D. S. Reeves, "MetaAware: Identifying metamorphic malware," in *Proc. Twenty-Third Annu. Comput. Secur. Appl. Conf. (ACSAC)*, Dec. 2007, pp. 411–420, doi: 10.1109/ACSAC.2007.9.

[12] C. Wang, D. Chen, L. Hao, X. Liu, Y. Zeng, J. Chen, and G. Zhang, "Pulmonary image classification based on inception-v3 transfer learning model," *IEEE Access*, vol. 7, pp. 146533–146541, 2019, doi: 10.1109/ACCESS.2019.2946000.

[13] S. Zeadally and M. Tsikerdekis, "Securing Internet of Things (IoT) with machine learning," *Int. J. Commun. Syst.*, vol. 33, no. 1, p. e4169, Jan. 2020, doi: 10.1002/dac.4169.

[14] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?" *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 41–49, Sep. 2018, doi: 10.1109/MSP.2018.2825478.

[15] H. HaddadPajouh, A. Dehghantanha, R. Khayami, and K.-K.-R. Choo, "A deep recurrent neural network based approach for Internet of Things malware threat hunting," *Future Gener. Comput. Syst.*, vol. 85, pp. 88–96, Aug. 2018, doi: 10.1016/j.future.2018.03.007.

[16] T.-L. Wan, T. Ban, S.-M. Cheng, Y.-T. Lee, B. Sun, R. Isawa, T. Takahashi, and D. Inoue, "Efficient detection and classification of Internet-of-Things malware based on byte sequences from executable files," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 262–275, 2020, doi: 10.1109/OJCS.2020.3033974.

[17] S. P. Choudhary and M. D. Vidyarthi, "A simple method for detection of metamorphic malware using dynamic analysis and text mining," *Proc. Comput. Sci.*, vol. 54, pp. 265–270, Jan. 2015, doi: 10.1016/j.procs.2015.06.031.

[18] D. Vasan, M. Alazab, S. Venkatraman, J. Akram, and Z. Qin, "MTHAEL: Cross-architecture IoT malware detection based on neural network advanced ensemble learning," *IEEE Trans. Comput.*, vol. 69, no. 11, pp. 1654–1667, Nov. 2020, doi: 10.1109/TC.2020.3015584.

[19] S. M. Sam, K. Kamardin, N. N. A. Sjarif, and N. Mohamed, "Offline signature verification using deep learning convolutional neural network (CNN) architectures GoogLeNet inception-v1 and inception-v3," *Proc. Comput. Sci.*, vol. 161, pp. 475–483, Jan. 2019, doi: 10.1016/j.procs.2019.11.147.

[20] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification," in *Proc. 8th Int. Symp. Visualizat. Cyber Secur.*, Jul. 2011, pp. 1–7, doi: 10.1145/2016904.2016908.

[21] T. G. Palla and S. Tayeb, "Intelligent mirai malware detection in IoT devices," in *Proc. IEEE World AI IoT Congr. (AIIoT)*, May 2021, pp. 0420–0426, doi: 10.1109/AIIoT52608.2021.9454215.

[22] U. Bayer, A. Moser, C. Kruegel, and E. Kirda, "Dynamic analysis of malicious code," *J. Comput. Virology*, vol. 2, no. 1, pp. 67–77, Aug. 2006, doi: 10.1007/s11416-006-0012-2.

[23] C. Willems, T. Holz, and F. Freiling, "Toward automated dynamic malware analysis using CWSandbox," *IEEE Secur. Privacy Mag.*, vol. 5, no. 2, pp. 32–39, Mar. 2007, doi: 10.1109/msp.2007.45.

[24] M. Rhode, P. Burnap, and K. Jones, "Early-stage malware prediction using recurrent neural networks," *Comput. Secur.*, vol. 77, pp. 578–594, Aug. 2018, doi: 10.1016/j.cose.2018.05.010.

[25] J. Saxe and K. Berlin, "Deep neural network based malware detection using two dimensional binary program features," in *Proc. 10th Int. Conf. Malicious Unwanted Softw. (MALWARE)*, Oct. 2015, pp. 11–20, doi: 10.1109/MALWARE.2015.7413680.

[26] B. Kolosnjaji, G. Eraisha, G. Webster, A. Zarras, and C. Eckert, "Empowering convolutional networks for malware classification and analysis," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 3838–3845, doi: 10.1109/IJCNN.2017.7966340.

[27] E. Grossi and M. Buscema, "Introduction to artificial neural networks," *Eur. J. Gastroenterology Hepatology*, vol. 19, no. 12, pp. 1046–1054, Dec. 2007, doi: 10.1097/meg.0b013e3282f198a0.

[28] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-VGG16 CNN model for big data places image recognition," in *Proc. IEEE 8th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2018, pp. 169–175, doi: 10.1109/CCWC.2018.8301729.

[29] I. Z. Mukti and D. Biswas, "Transfer learning based plant diseases detection using ResNet50," in *Proc. 4th Int. Conf. Electr. Inf. Commun. Technol. (EICT)*, Dec. 2019, pp. 1–6, doi: 10.1109/EICT48899.2019.9068805.

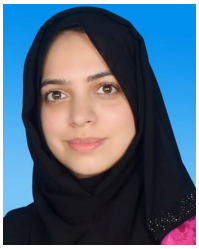
[30] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114. [Online]. Available: http://proceedings.mlr.press/v97/tan19a.html

[31] H. Ide and T. Kurita, "Improvement of learning for CNN with ReLU activation by sparse regularization," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2684–2691, doi: 10.1109/IJCNN.2017.7966185.

[32] S. V. Dicholkar and D. Sekhar, "Review-IoT security research opportunities," in *Proc. Int. Conf. Converg. Digit. World-Quo Vadis (ICCDW)*, Feb. 2020, pp. 1–4, doi: 10.1109/ICCDW45521.2020.9318641.

[33] Q.-G. Lin, N. Li, Q. Qi, and J.-B. Hu, "Classification of IoT malware based on convolutional neural network," in *Proc. Int. Conf. Service Sci. (ICSS)*, Aug. 2020, pp. 51–57, doi: 10.1109/ICSS50103.2020.00016.

[34] R. J. Maulana and G. P. Kusuma, "Malware classification based on system call sequences using deep learning," *Adv. Sci., Technol. Eng. Syst. J.*, vol. 5, no. 4, pp. 207–216, Jul. 2020, doi: 10.25046/aj050426.
[35] Q. Li, J. Mi, W. Li, J. Wang, and M. Cheng, "CNN-based malware variants detection method for Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 23, pp. 16946–16962, Dec. 2021, doi: 10.1109/JIOT.2021.3075694.
[36] F. Ullah, H. Naeem, S. Jabbar, S. Khalid, M. A. Latif, F. Al-turjman, and L. Mostarda, "Cyber security threats detection in Internet of Things using deep learning approach," *IEEE Access*, vol. 7, pp. 124379–124389, 2019, doi: 10.1109/ACCESS.2019.2937347.



FAIZA HABIB received the B.S. degree in computer science from Abbottabad University of Science and Technology and the M.S. degree in computer from Hazara University Mansehra. She was a Lecturer with Women University Sawabi. She is currently a Lecturer with Abasyn University, Islamabad Campus. Her research interests include machine learning and deep learning.

SYED HAMAD SHIRAZI received the Ph.D. degree in computer science from Hazara University Mansehra, Pakistan, in 2017. He is currently an Assistant Professor with the Department of Computer Science and Information Technology, Hazara University Mansehra. He serves as the Principal Investigator for the HEC (NRPU) funded project titled "Disease Prediction Support System for Anemic RBC Based on Machine Learning (DPSS-ARBC)," a collaborative effort with the Shaukat Khanam Cancer Hospital. Additionally, he is actively involved in the ongoing project, "Remote Sensing and Geospatial Analysis for Modeling and Predicting the Impacts of Climate Change on Glacier Water Resources and Crop Mapping in Pakistan." His commitment to academic excellence is reflected in his extensive publication record, boasting more than 40 articles in reputable international journals and conferences. His research interests include advancing the field of computer science, including computer vision, texture analysis, object recognition, pattern recognition, medical imaging, machine learning, deep learning, and information security. Recognized for his dedication to education and research, he received the Best Teacher Award from the Government of Khyber Pakhtunkhwa, in 2018, and the Research Productivity Award from COMSAT University Islamabad, in 2014.

KHURSHEED AURANGZEB (Senior Member, IEEE) received the B.S. degree in computer engineering from the COMSATS Institute of Information Technology Abbottabad, Pakistan, in 2006, the M.S. degree in electrical engineering (system on chip design) from Linkoping University, Sweden, in 2009, and the Ph.D. degree in electronics design from Mid Sweden University, Sweden, in June 2013. He is currently an Associate Professor with the Department of Computer Engineering, College of Computer and Information Sciences, King Saud University (KSU), Riyadh, Saudi Arabia. He has authored and coauthored more than 90 publications, including IEEE/ACM/Springer/Hindawi/MDPI journals and flagship conference papers. He has obtained more than 15 years of excellent experience as an instructor and a researcher in data analytics, machine/deep learning, signal processing, electronics circuits/systems, and embedded systems. He has been involved in many research projects as a principal investigator and a co-principal investigator. His research interests include embedded systems, computer architecture, signal processing, wireless sensor networks, communication, and camera-based sensor networks, with an emphasis on big data and machine/deep learning with applications in smart grids, precision agriculture, and healthcare.

ASFANDYAR KHAN received the M.S. and Ph.D. degrees in computer science from Hazara University, Mansehra, Pakistan, in 2015 and 2023, respectively. He is currently a Senior Lecturer with the Department of Computer Science and Information Technology, Hazara University Mansehra, Pakistan. His research interests include smart grid network design, planning, electricity consumption, resource handling and allocation, wireless body area networks, the IoT, cloud computing, and machine learning.

BHARAT BHUSHAN (Senior Member, IEEE) received the B.Tech. degree (Hons.) in computer science and engineering, the M.Tech. degree (Hons.) in information security, and the Ph.D. degree in computer science and engineering from the Birla Institute of Technology, Mesra, India, in 2012, 2015, and 2021, respectively. He is currently an Assistant Professor with the Department of Computer Science and Engineering (CSE), School of Engineering and Technology, Sharda University, Greater Noida, India. In the past, he was an Assistant Professor with the HMR Institute of Technology and Management, New Delhi, and an Network Engineer with HCL Infosystems Ltd., Noida. In the year 2021 and 2022, he was with Stanford University, USA, listed Dr. Bharat Bhushan in the top 2% scientists list. He earned numerous international certifications, such as CCNA, MCTS, MCITP, RHCE, and CCNP. He has published more than 150 research papers in various renowned international conferences and SCI indexed journals, including *Journal of Network and Computer Applications* (Elsevier), *Wireless Networks* (Springer), *Wireless Personal Communications* (Springer), *Sustainable Cities and Society* (Elsevier), and *Emerging Transactions on Telecommunications* (Wiley). He has contributed with more than 30 book chapters in various books and has edited 20 books from the most famed publishers, such as Elsevier, Springer, Wiley, IOP Press, IGI Global, and CRC Press. He is a Series Editor of two prestigious Scopus indexed book series named *Computational Methods for Industrial Applications* and *Future Generation Information System* (CRC Press and Taylor and Francis, USA). He has served as keynote speaker (resource person) in numerous reputed faculty development programs and international conferences held in different countries, including India, Iraq, Morocco, China, Belgium, and Bangladesh. He has served as a reviewer/an editorial board member for several reputed international journals. He is a member of numerous renowned bodies, including IAENG, CSTA, SCIEI, IAE, and UACEE.

MUSAED ALHUSSEIN received the B.S. degree in computer engineering from King Saud University, Riyadh, in 1988, and the M.S. and Ph.D. degrees in computer science and engineering from the University of South Florida, Tampa, FL, USA, in 1992 and 1997, respectively. Since 1997, he has been a Faculty Member of the Computer Engineering Department, College of Computer and Information Science, King Saud University (KSU), Riyadh, Saudi Arabia, where he is currently a Professor. Recently, he has been successful in winning a research project in the area of AI for healthcare, which is funded by the Ministry of Education, Saudi Arabia. He is also the Founder and the Director of the Embedded Computing and Signal Processing Research (ECASP) Laboratory. His research interests include typical computer architecture and signal processing topics, with an emphasis on big data, machine/deep learning, VLSI testing and verification, embedded and pervasive computing, cyber-physical systems, mobile cloud computing, big data, eHealthcare, and body area networks.