Mehran University Research Journal of Engineering and Technology
Vol. 41, No. 1, 80 - 91, January 2022
p-ISSN: 0254-7821, e-ISSN: 2413-7219
DOI: https://doi.org/10.22581/muet1982.2201.08

Malware Detection and Classification in IoT Network using ANN

Ayesha Jamal[1a], Muhammad Faisal Hayat[1b], Muhammad Nasir[1c]

RECEIVED ON 10.09.2020, ACCEPTED ON 05.05.2021

ABSTRACT

Internet of Things is an emerging technology in the modern world and its network is expanding constantly. Meanwhile, IoT devices are a soft target and vulnerable to attackers. The battle between malware attackers and security analysts is persistent and everlasting. Because malware is evolving constantly and thus asserting pressure on researchers and security analysts to cope up with modern threats by improving their defense systems. Complexity and diversity of current malicious software present immense challenges for protecting IoT networks from malware attacks. In this paper, we have explored the potential of neural networks for detection and classification of malware using IoT network dataset comprising of total 4,61,043 records with 3,00,000 as benign while 1,61,043 as malicious. With the proposed methodology, malware is detected with an accuracy of 94.17% while classified with 97.08% accuracy.

Keywords: Internet of Things (IoT), Malware Detection, Malware Classification, Artificial Neural Network (ANN), Artificial Intelligence (AI).

1. INTRODUCTION

In modern times, Internet of Things (IoT) is an interrelated network of multiple devices in which data is automatically collected from the environment by the sensors, transferred over the internet without human help and intervention such as home appliances, traffic lights, and lamp posts *etc*., that are related to the Internet.

IoT devices have a range of sensors that render useful data generation without human-to-human or human-to-machine interaction [1]. The Internet of Things is known as the third industrial transition. It is known as "interconnection, through the Internet, from computer equipment embedded in everyday objects, allowing it to send and receive data" [2]. The market for IoT is growing at a spectacular rate, beginning with 2 billion artifacts in 2006, a forecast rise of 200 billion by 2020 out of 200%. IoT sensors or appliances also gather and process temporal and spatial statistics for unique incidents and surroundings tackling specific challenges. IoT is seen in most fields: home, school, culture, energy distribution, finance, healthcare, tourism, smart cities, and also for transport. The objects of IoT are getting cleverer, diagnosis is smarter and interactions are becoming instructive [3].

1.1 Security Challenges in IoT Network

The Internet of Things (IoT) [4] is a sensing network suitable for wired/wireless devices with limited resources as shown in Fig 1. IoT apps are progressively targeted by malware-using attackers easier to infect computers than traditional ones. That is because of several reasons [5], for example, the existence of legacy devices, no security upgrades, low safety goals inside the cycle of development, weak login credentials, *etc*. As per Kaspersky Lab [6], in 2016 most of the IoT devices inspected were

[1] Department of Computer Engineering, University of Engineering and Technology Lahore, Pakistan.
Email: ayeshajamal27@gmail.com (Corresponding Author), muhammad.faisal.hayat@uet.edu.pk, nasirmaharvi95@gmail.com

unreliable as in these devices had either default secret key or unpatched vulnerabilities. As it were, these devices can be effectively traded off utilizing malware, for example, Hajime and Mirai [7].

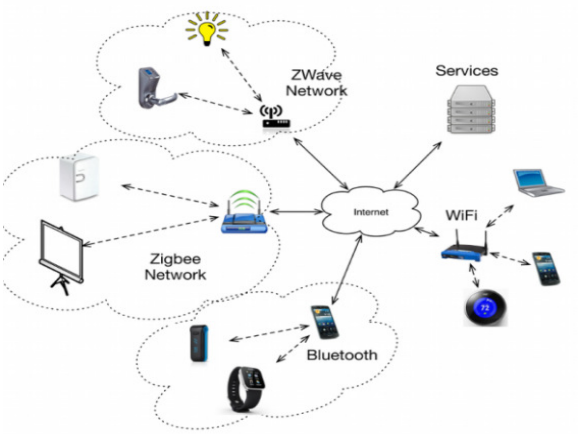

Fig. 1: IoT Network Architecture

IoT devices are a soft target for hackers or unauthorized users as they are simpler to taint than regular PCs for the following reasons [8]:

- Numerous IoT devices are associated with the Internet without any updates in security.
- For development of IoT devices, security is given a low priority.
- Implementing cryptography techniques in IoT devices is computationally costly because of memory and power limitations.
- Login credentials that are either given by the user or by the manufacturer are weak in IoT devices.
- Sometimes few backdoors are left by vendors of IoT devices to provide remote support for that device.
- IoT devices are often associated with the Internet without experiencing a firewall.

IoT software manufacturers don't routinely upgrade their apps unless the user initiates firmware updates. Due to resource constraints [9], these systems cannot run full-fledged protection protocols, so IoT devices are vulnerable to attack for longer periods (e.g. their default login keys, unpatched bugs) [10].

IoT devices operate more in an unattended environment, so there is a reasonable risk that an attacker can gain physical access to them intentionally. As a result, attackers can obtain valuable information via the communication channel by listening to the conversation secretly, since most IoT devices use wireless links. These devices do not integrate strong security features because there are restricted computing and power resources [11]. The implementation of strong security mechanisms is not only difficult due to the limited available resources but also due to non-trustful contact with the environment. Given the likelihood of compromised IoT devices in an IoT network, a comprehensive protection approach must be established based on time-to-time patching of vulnerabilities [12].

In recent years, numerous methods have been proposed by many researchers regarding malware detection and classification using machine learning algorithms. These works mainly focus on malware detection in Android devices, Windows or OS malware, and limited work on malware identification in IoT network which is a substantial security threat in recent times. Based on the above discussion, there is a need for an efficient technique that generates the best possible results for malware detection and classification in a shorter time.

In this paper challenge of detection and classification of malware using network traffic analysis has been taken up. Main contributions of the paper are summarized as:

- Proposed the first ANN to detect malware by analyzing packets of network traffic generated by the IoT network.
- Another ANN is proposed that classifies malware families based on network traffic behavior.
- The proposed methodology is compared with traditional ML algorithms i.e., k-NN and Naïve Bayes.
- Analysis of results depicts that the proposed methodology is efficient for detecting malware with an accuracy of 94.17% while classifying malware with an accuracy of 97.08%.

This paper is organized in different sections. Section 2 presents an overview of past literature. Section 3 demonstrates related background information. Dataset description and creation is explained in Section 4

while experimental results are evaluated in Section 5. In the end conclusion along with future work and comparison is given in Section 6 and 7 respectively.

2. LITERATURE SURVEY

There are several works in the literature related to malware analysis, detection, and classification. Intrusion or malware detection is a trending area of research. However, it is unlikely that the resource-constrained existence of most IoT devices and customized operating systems, traditional malware detection and prevention solutions would fit the real world. Malware can exploit vulnerabilities in compromised IoT systems, or it can cause specific limitations on some IoT apps. Therefore, the IoT network's security requirement that needs to be addressed is fixing malware.

Liu *et al*. [8] presented a multi-layer learning framework for classification of malware by converting samples to greyscale images. Machine learning algorithms *i.e*., the k- Nearest Neighbour (k-NN) and Random Forest (RF) are applied on malware datasets, compared with existing work, and accuracy is improved.

Kumar and Lim [6] presented a solution for the detection of malware in large scale networks rather than detection based on hosts. ML technique is used to analyze traffic patterns for detection of malware activity in IoT devices, store those traffic patterns in the database and perform necessary countermeasures for detecting the malicious activity of IoT bots i.e., blocking of traffic generated by botnets and report to network administrators. Target is to identify IoT bots before the actual attack i.e., in the scanning phase. Past work is done on PC based bots rather than IoT bots. Features extracted are the number of unique IP addresses and the number of packets sent to a single IP address. The dataset includes malware scripts generated based on publicly available exploits. Using Wireshark packet information is gathered. Feature vectors were developed having two classes as malicious and benign. Comparison of machine learning algorithms i.e., k-NN, Random Forest, and Gaussian Naïve Bayes is made. 94.44% accuracy achieved using k-NN.

Kumar *et al.* [13] proposed a framework for classification of benign (6192) and malware (5560) apps using ML techniques and block-chain. Features extracted are permissions, environmental information, etc. extracted from ML techniques are stored in block-chain. 98% accuracy achieved using Naïve Bayes Algorithm.

Koroniotis *et al.* [14] proposed a framework for the detection of infected and benign traffic from both IoT and non-IoT devices. Dataset used is BoT-IoT. Support Vector Machine (SVM), Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) techniques are used. The highest accuracy is achieved using SVM.

Alasmary *et al*. [15] proposed a control flow graph methodology for detection, classification, and comparison of malware in IoT and Android applications. IoT (2962) and Android (2891) malware samples are collected from different resources that are analyzed. Features extracted were nodes count, edges count, shortest path, *etc*. using Control Flow Graph (CFG). A comparison of different algorithms *i.e*., Covolutional Neural Network (CNN), RF, Logistic Regression (LR), and SVM was made. CNN gave a detection accuracy of 99.66% while for classification as 99.32%.

Nguyen *et al*. [16] proposed an approach for detecting IoT botnet using Printable String Information (PSI) graphs. Dataset consists of 11200 elf files, 7199 malware samples while 4001 benign samples. Function call graphs were created using these samples. Further PSI graphs were created using functions that were close to IoT botnets. CNN classifies malware and benign samples for IoT devices using feature vector data from PSI graphs that indicate the rate and direction of change in features. 98.7% accuracy achieved using the PSI graph-based approach. However, there are also some limitations in this work as it has an analysis of control flow graphs that are complex, effort, and time-consuming.

Vinayakumar *et al.* [17] proposed framework ScaleMalNet that can handle Big Data of malicious samples. This paper also contributes to presenting novel image processing techniques for the classification of malware. Different deep and machine

learning models are analyzed. Publically available dataset Ember is used for performance analysis of the proposed framework that consists of 70,140 benign and 69,860 malicious samples. Classifiers of Machine Learning (ML) *i.e*., k-NN, Naïve Bayes, Decision Tree, Ada Boost, Logistic Regression, RF, SVM, and Deep Neural Network (DNN) are applied to the dataset. DNN outer-performs classical ML algorithms with an accuracy of 98.9%. For classification, DeepImageMalDetect *i.e*., combination of deep learning models based on image processing technique along with LSTM and CNN is proposed. Malimg dataset along with privately collected samples is used for malware family classification. The malicious dataset consists of 9339 samples having 2 different families. The hybrid approach has achieved an accuracy of 96.3% in the classification of malware. Unluckily, related works in the past are not vigorous because of the limited number of samples of data.

Yin *et al.* [18] proposed mechanism for dynamic analysis of malware using a deep neural network that comprises of three modules: one that monitors and analyze the dynamic behavior of malware, second that processes log files generated by previous module and third that consists of deep neural network mainly CNN used to detect and classify malware. Dataset comprises of 10,000 malware samples from 5 families each has 2000 samples. 97.3% of accuracy is achieved. In this work, data samples are a less and inadequate set of malware families is used.

Aman *et al*. [19] proposes a novel framework that classifies and identifies malware samples. Dataset comprises of 20 families of malware each has 2000 samples. 32,475 malware samples are extracted including 9 families. VirusTotal is used to assign labels to malware families and features are extracted. 67% dataset is used for training while 33% is used for testing the model. J48, Naïve Bayes, and Random Forest are applied to the dataset. Random Forest achieves better performance *i.e*., 0.9914 AUC in comparison with J48 and Naïve Bayes.

David and Netanyahu [20] proposed technique Deepsign that detects malware automatically using the process of creating signatures. Dataset is created on basis of entries of the registry, Application Programming Interface (API) calls behavioral logs, port accesses, searches of the web, *etc*. in the sandbox, and logs are then converted to binary vector. A deep belief network is used for malware classification and accuracy of 98.6% is achieved. More or less, there exists a series of limitations in existing literature in terms of data samples, features extracted, *etc*. A cyber-security research survey was conducted by Buczak and Guven [21] using machine learning algorithms and data mining techniques. Their survey affirmed that there is lack of a labelled dataset that creates a critical gap in the literature that must be addressed in order to develop a promising anomaly-based intrusion detection method.

As we look into related work, researchers detect and classify malware on windows, android apps, and IoT devices which comprise of binary image, control flow graphs, and portable executable files while using network packets very little or no work is seen. Moreover, data samples used in literature are small including less benign and malware samples as well as least malware families in terms of classification. Koroniotis *et al.* [22] and Hamza *et al*. [23] recently proposed network-based IoT datasets that are comprised of attack scenarios. However, the datasets did not have a variety of attack types such as ransomware and Cross-Site Scripting (XSS) nor they contain sensor measurement data of IoT devices.

Traditional machine learning-based malware detection and classification rely on feature engineering, feature learning, and feature representation techniques that require extensive domain-level knowledge. In contrast to ML algorithms, the neural network tries to learn features from data in an incremental manner. So, there is a need for a methodology that can efficiently detect and classify malware in IoT networks using network traffic analysis.

These issues have persuaded us to come up with an IoT-related dataset that contains sensors' reading data as an information source for data-driven IoT-based Intrusion Detection System (IDS) to properly monitor the internal behavior of IoT applications, hereby securing them from malicious activities.

3. THEORETICAL BACKGROUND

This Section presents an overview of malware as a

security threat for IoT network, analysis, and detection techniques of malware, also an overview of machine learning and deep learning approaches.

3.1 Malware

Malware is defined as software that fulfills the harmful intent of an attacker. Different researchers define malware with different definitions like a code that is added to the system to deliberately cause damage or invert the actual task of the system. Malware is of various kinds like Trojan horse, virus, worm, *etc*. as shown in Fig 2. Trojan horse is a kind of malware that is planted in a system or app by its manufacturer. The system performs intended actions but it also performs some invalid actions. A virus is a program that spreads to other programs by replication. An infected program that causes harm to other programs is called the host of the virus. The host spreads itself to other system programs. The worm is a program that spreads to other programs by replication of its code execution. The difference between worm and virus is that the former needs host to cause damage. The worm spreads and tries to infect the whole network [24].

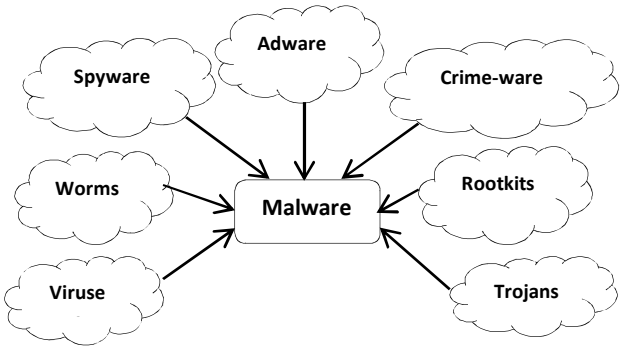

Fig. 2: Different types of Malware

The destruction caused by malware has increased adequately within the prior years. The main reason is the expanding recognition of the Internet and at the same time, there is an increase within the wide variety of vulnerable machines available because of security negligent users.

The destruction caused by malware has increased adequately within the prior years. The main reason is the expanding recognition of the Internet and at the same time, there is an increase within the wide variety of vulnerable machines available because of security negligent users. Another cause is the sophistication of malicious software has been improved over time. Malicious software is based on signatures. If signatures are identified in a program's code that is asserted as malware then it can easily be detected [25].

3.2 Evolution of Malware

Dangers from malware are not new, even though malware or digital danger chasing stays a continuous challenge. For instance, with the expanding prominence of IoT devices and the absence of security insurance for such devices, these devices can be powerless against malware assaults [26].

Malware is deliberately intended to harm a PC, server, or any system and it has become one of the most noteworthy dangers on the Internet. It may have different names like virus, Trojan, ransomware, worm, command and control bot, *etc*. [17]. With the assistance of modern tools, it turns out to be easy to create new malware, bringing about an exceptionally fast increment in the quantity of malware. Moreover, those new malicious codes have the same behavior as benign codes making them harder to be distinguished, which have represented a noteworthy challenge to the vendors of anti-virus [27].

Early day malware was not encrypted utilizing complex cipher techniques and therefore were effectively identified and arranged by cross-coordinating some bit of code. But with the ongoing ideas of polymorphism and transformative nature like jumbling, malware characterization [14] turns into a difficult and dreary undertaking. Polymorphic malware exploit is an encryption technique, which encodes the code each time it repeats, while the encryption key stays steady which makes it simpler to identify. In the examination, metamorphic malware which not just encodes the code each time it repeats at the same time additionally changes its encryption key, which makes it difficult to recognize [28].

3.3 Malware Analysis

Machine is analyzed to comprehend the behavior and their contents. Malware analysis is the procedure of making sense of the ability of malware and answers to the following queries *i.e*., how malware functions,

which machines and projects are influenced, which realities are being harmed and taken, and so forth. Malware can be analyzed either by examination of its code or by creating a safe environment for its execution. There are specifically two main strategies to investigate malware:

- Static
- Dynamic

Static analysis inspects the malware without executing the genuine code [25]. The patterns of detection used in static analysis comprise of n-grams, string signature, syntactic library call, control drift graph, and opcode frequency distribution, etc. For static analysis, the executable has to be decrypted.

On the elective hand, dynamic assessment inspects the malware practices while executing its code in a safe and controlled environment *i.e*., installing different software like Wireshark, Regshot, Capture BAT, *etc*. Malware assessment begins with fundamental static assessment and gets done with cutting edge dynamic assessment.

In comparison with static analysis, dynamic analysis is far better and does not need the executables to be disassembled. It unveils the natural behavior of malware that is more volatile to static analysis. The digital surroundings in which malware are finished are not like the actual one and the malware may perform in distinct approaches resulting in artificial conduct as an alternative than the exact one [29].

3.4 Malware Detection Techniques

Daily usage of the Internet comes with both its pros and cons. Internet world crimes are growing faster as compared to real-world crime because of different cyber-attacks infected with modern malware that can bypass all security measures. In the preceding days, the malware was simple and easy to detect but in the modern days, it is more complicated and difficult to detect. The signature-based approach was used before for malware detection but that is an old methodology and cannot detect modern malware that is complicated [30]. New methods have also been proposed for malware detection still it's impossible to detect all new malware. Malware detection involves three stages: first is to analyze malware, second is to extract features and third is to classify malware and benign. Malware detection can be static as well as dynamic i.e., can be detected when code is not running as well as detected when code is running. Different approaches for malware detection are described below [31]:

- Signature-based detection
- Behavior-based detection
- Heuristic-based detection
- Model-checking based
- Deep learning-based detection
- Cloud-based detection
- Mobile-based detection
- IoT-based detection

3.5 Machine Learning

Machine learning is the branch of Artificial Intelligence (AI) that can function automatically and learn from the previous and new experiences without being explicitly programmed or any human interaction. Machine learning approaches can be used to classify data automatically. This approach is further categorized as supervised learning and unsupervised learning. The difference between these two approaches is that in a supervised learning approach, the class label is present in the data before we apply any learning algorithm [32]. And in an unsupervised approach, the class label is not present so the learning algorithm has to analyze data and assign a class to it by organizing similarity clusters or groups.

3.6 Artificial Neural Network

Artificial Neural Network [33] is a network of numerous small connecting elements known as neurons also called the perceptron. ANN works on the principle of human brain. Each neuron can make decisions and information is transferred to other connected neurons that are organized in layers. It works as an artificial human nervous system that is used for transmitting, processing, and receiving information. A type of artificial neural network in which there exists one input layer for input variables, one hidden layer, and one output layer is known as the Shallow Neural Network. ANN with more than one hidden layer of neurons that process the inputs is known as Deep Neural Network. In ANN there are three layers which are as follows:

Input Layer (All inputs provided to the model through this layer)

- Hidden layer (maybe more than one depending upon the problems and used for processing the inputs received from input layer)
- Output layer (For prediction)

4. DATASET DESCRIPTION AND CREATION

Dataset used in this paper is the ToN_IoT dataset [34] that is collected from the University of New South Wales (UNSW), Canberra created at their IoT Lab by Dr. Nour Mustafa. Dataset is called ToN_IoT as it consists of Telemetry datasets of IIoT and IoT sensors, datasets of Operating systems for both Ubuntu/Windows and datasets of Network traffic.

Current security solutions, including threat hunting and intelligence, digital forensics, malware detection, and intrusion detection are trending research areas in the domain of cybersecurity. With the advancement in AI, particularly deep learning, current solutions for security makes use of AI models yet these are not reliable due to diverse variety and complexity of recent hacking categories, unavailability of data sources for training, and validation of AI models. To fulfill that gap, a new dataset named ToN_IoT is designed to evaluate the fidelity of current security solutions based on AI models. Testbed developed consists of three tiers:

- Edge (IoT and Network devices)
- Fog (VM's and gateways)
- Cloud (cloud services linked with fog and edge tiers including visualization and data analytics)

Dataset is collected in pcap format using Wireshark that is converted to csv format. Dataset consists of both normal and attack scenarios. Tools used in testbed are Security Onion, Kali Linux, Wireshark, and Bro (named as Zeek).

4.1 Statistics of Dataset

TON IoT original dataset contains more than 22M ToN_IoT original dataset contains more than 22M records. For training and testing purposes original dataset is filtered to generate standard features and their labels.

The training and testing dataset consists a total of 4,61,043 records (as shown in Table 1) with 3,00,000 as normal or benign while 1,61,043 as malware that can be visualized in Fig. 3.

Table 1: ToN_IoT Dataset Statistics

Type	No. of Records
benign	3,00,000
backdoor	20,000
ddos	20,000
dos	20,000
injection	20,000
mitm	1043
password	20,000
ransomware	20,000
scanning	20,000
xss	20,000

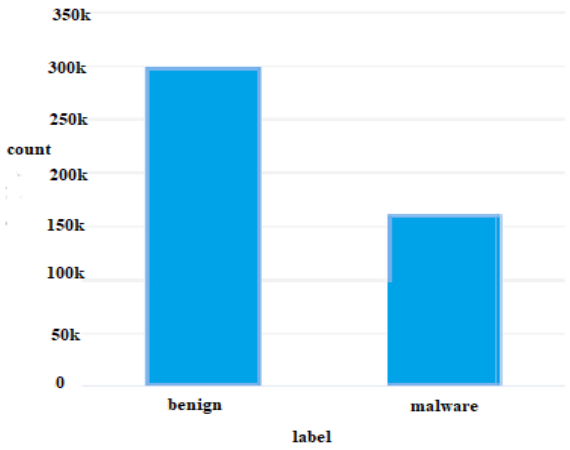

Fig. 3: Dataset Statistics

4.2 Features of Dataset

44 important features are extracted from the dataset along with labels and type. Some of them are described in Table 2.

4.3 Malware Families

Malware data consists of 9 attacking families (as shown in Fig. 4) listed below:

- Scanning Attack
- Denial of Service (DoS) Attack
- Distributed Denial of Service (DDoS) Attack
- Ransomware Attack

- Backdoor Attack
- Injection Attack
- XSS Attack
- Password Attack
- Man in the Middle (MITM) Attack

Table 2: Features Extracted from Dataset

Feature	Description
ts	Timestamp of connection
src_ip	Source IP address
src_port	Source port
dst_ip	Destination IP address
dst_port	Destination port
proto	Protocols of the transport layer
service	Dynamically detected protocols
duration	Time of packet connections *i.e.,* subtracting "time of last seen packets" and "time of first seen packets"
src_bytes	Source bytes originated from payload bytes
dst_bytes	Destination bytes originated from payload bytes
conn_state	Connection states
missed_bytes	Number of missing bytes
src_pkts	Number of original packets estimated from source
src_ip_bytes	Number of original IP bytes which is the total length of IP header field of source
dst_pkts	Number of destination packets estimated from destination
dst_ip_bytes	Number of destination IP bytes which is the total length of IP header field of destination

5. EXPERIMENTAL SETUP

In the experimental setup, we have proposed two ANN's based on feed-forward and backpropagation architecture that is build using different Python libraries i.e., Pandas, Tensor Flow *etc*. For malware detection, ANN consists of an input layer, three hidden layers consist of 150, 300, 150 neurons respectively while the output layer consists of one neuron as it is binary classification *i.e*. 0 or 1. Input and hidden layers have ReLU as activation function and sigmoid is used for the output layer. The loss function used for malware detection is Binary Cross Entropy that is for binary classification. Adam which is gradient descent optimizer is used as an optimizer for that loss function. Dataset is split into train and test data in the ratio of 70% and 30%. Trained data is executed for 60 epochs having a batch size of 80. The model achieved an accuracy of 94.17% as depicted in the confusion matrix shown in Fig 5.

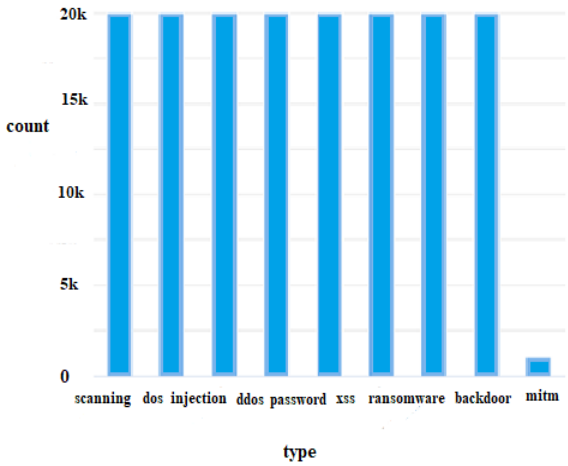

Fig.4: Malware Family Statistics

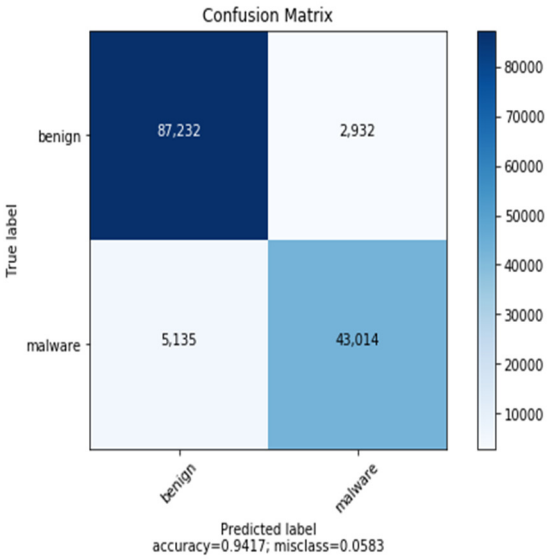

Fig.5: Confusion Matrix for Malware Detection

For malware classification, ANN consists of an input layer, three hidden layers consist of 150, 70, 100 neurons respectively while the output layer consists of 9 neurons as it is a multiclass classification having nine malware families i.e., from 0 to 8. The output layer has 9 neurons because it is fed with 9 arrays. Input and hidden layers have ReLU as activation function and softmax is used for output layer as for multiclass classification softmax is required. The loss

function used for malware detection is Categorical Cross Entropy that is for multiclass classification. Adam which is gradient descent optimizer is used as an optimizer for loss function. The dataset used for malware family classification consists of 1,61,043 records. Dataset is split into train and test data in the ratio of 70% and 30%. Trained data is executed for 100 epochs having a batch size of 70. The model achieved an accuracy of 97.08% as depicted in the confusion matrix shown in Fig 6.

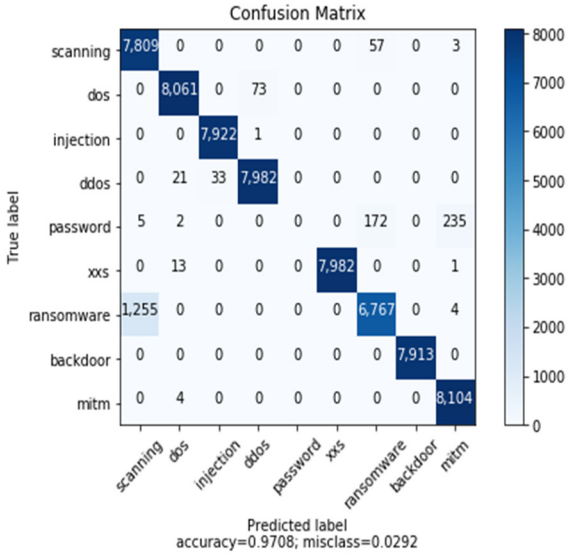

Fig.6: Confusion Matrix for Malware Classification

6. EVALUATION MEASURES AND RESULTS

Common performance indicators for evaluating the performance of classifiers are:

- True Positive (TP): Ration of benign samples classified as benign.
- True Negative (TN): Ration of malware samples classified as malware.
- False Positive (FP): Ratio of malware classified as benign.
- **False Negative (FN):** Ratio of benign classified as malware.
- **Accuracy:** Ratio of correctly predicted observations to total observations.

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

- **Precision:**

The ratio of correctly predicted positive observations and the total predicted positive observations.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- **Recall:**

The ratio of correctly predicted positive observations and the total predicted observations of the actual class.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- **F1 Score:**

The weighted average of precision and recall.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Evaluation measures for malware detection along with classification report (**Table 3**) are as follows:

$$\text{Accuracy} = \frac{89638 + 43014}{87232 + 2932 + 43014 + 5135}$$

$$\text{Accuracy} = 0.9417$$

Table 3: Classification Report for Malware Detection

class	precision	recall	f1-score	support
0	0.94	0.97	0.96	90164
1	0.94	0.89	0.91	48149

Evaluation measures for malware detection along with classification report (as in Table 4) are as follows:

$$\text{Accuracy} = 0.9708$$

Accuracy graphs of neural networks for malware detection and classification are shown in Fig 7 and Fig. 8 respectively.

Table 4: Classification Report for malware classification

class	precision	recall	f1-score	support
0	0.86	0.99	0.92	7869
1	1.00	0.99	0.99	8134
2	1.00	1.00	1.00	7923
3	0.99	0.99	0.99	8036
4	0.00	0.00	0.00	414
5	1.00	1.00	1.00	7994
6	0.97	0.84	0.90	8026
7	1.00	1.00	1.00	7913
8	0.97	1.00	0.98	8108

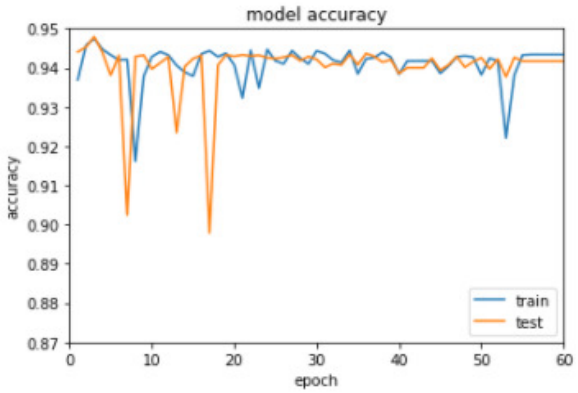

Fig.7: Accuracy of the model against unknown malware detection

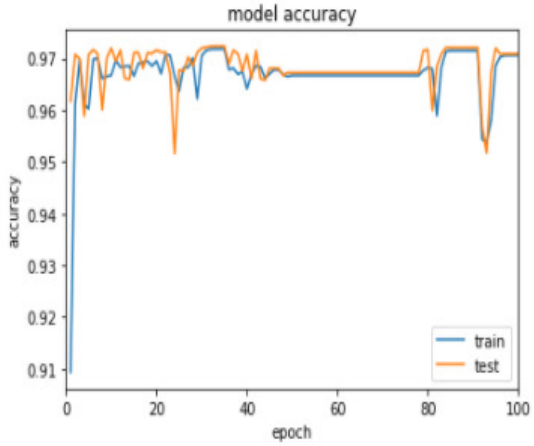

Fig.8: Accuracy of the model against unknown malware classification

A comparison of proposed algorithm is made with k-NN and Naïve Bayes as shown in Table 5 which depicts that ANN outer-performs classical ML algorithms.

Table 5: Comparison of ANN with ML algorithms

Algorithm	Detection Accuracy	Classification Accuracy
k-NN	0.8824	0.8764
NB	0.7639	0.8295
ANN	0.9417	0.9708

7. CONCLUSION & FUTURE WORK

As the diversity and range of IoT devices are promptly expanding, it is critical to secure such devices in the network against vulnerable attacks i.e., malware. We have highlighted security challenges in IoT network, background related to malware evolution, analysis, detection techniques, and different approaches. Various network datasets, for example, KDDCUP99, NSL-KDD [35], UNSW-NB15 [36] were generated for evaluating IDSs; however, they do not include any specific characteristics of IoT applications as these datasets contain neither sensors' reading data nor IoT network traffic.

Most of the recently published datasets [22, 23, 35, 36] are network-based datasets, which primarily contain packet-level and flow-level information or a combination of both, for detecting attacks on the IoT network. However, they do not have the actual data generated from sensor readings.

This paper fills the gap of the unavailability of the dataset that contains a variety of network attacks as well as a real-world network dataset. In comparison with the literature [13-15, 17], proposed methodology is highly capable to discriminate between malware and benign samples with an accuracy of 94.17% as well as classify malware families with an accuracy of 97.08% on basis of network traffic generated by IoT network.

Future research involves the construction of next-generation firewalls that can act as an intermediary between external networks and IoT networks preventing direct contact between two. Examine and identify advanced malware will also be taken into account in the future.

ACKNOWLEDGEMENT

Authors are grateful to Department of Computer Engineering, University of Engineering & Technology Lahore, Pakistan, to conduct this study.

REFERENCES

1. Guan Z. Li J., Wu L., Zhang Y., Wu J., Du X., "Achieving efficient and secure data acquisition for cloud-supported internet of things in smart grid", *IEEE Internet of Things Journal,* Vol. 4, pp. 1934-1944, 2017.
2. Hussain F., Hussain R., Hassan S. A., Hossain E., "Machine learning in IoT security: current solutions and future challenges," *IEEE Communications Surveys and Tutorials,* 2020.

3. Chaabouni N., Mosbah M., Zemmari A., Sauvignac C., and Faruki P., "Network intrusion detection for IoT security based on learning techniques", *IEEE Communications Surveys & Tutorials,* Vol. 21, pp. 2671-2701, 2019.
4. Al-Fuqaha A., Guizani M., Mohammadi M., Aledhari M., Ayyash M., "Internet of things: A survey on enabling technologies, protocols, and applications", *IEEE Communications Surveys & Tutorials,* Vol. 17, pp. 2347-2376, 2015.
5. Yang Y., Wu L., Yin G., Li L., Zhao H., "A survey on security and privacy issues in Internet-of-Things," *IEEE Internet of Things Journal,* Vol. 4, pp. 1250-1258, 2017.
6. Kumar A., Lim T. J., "Edima: early detection of IoT malware network activity using machine learning techniques", *Proceedings of the 5th World Forum on Internet of Things (WF-IoT)*, pp. 289-294, Limerick, Ireland, 15-18 April 2019.
7. Kumar A., Lim T. J., "Early Detection Of Mirai-Like IoT Bots In Large-Scale Networks Through Sub-Sampled Packet Traffic Analysis". In *Advances in Information and Communication, Lecture Notes in Networks and Systems,* Vol. 70, Springer, 2020.
8. Liu Y.-S., Lai Y.-K., Wang Z.-H., and Yan H.-B., "A new learning approach to malware classification using discriminative feature extraction", *IEEE Access,* Vol. 7, pp. 13015-13023, 2019.
9. Alkhalil A., Ramadan R. A., "IoT data provenance implementation challenges", *Procedia Computer Science,* Vol. 109, pp. 1134-1139, 2017.
10. Roman R., Zhou J., Lopez J., "On the features and challenges of security and privacy in distributed internet of things", *Computer Networks,* Vol. 57, pp. 2266-2279, 2013.
11. Al-Garadi M. A., Mohamed A., Al-Ali A., Du X., Ali I., Guizani M., "A survey of machine and deep learning methods for internet of things (IoT) security", *IEEE Communications Surveys and Tutorials,* 2020.
12. Miettinen M., Marchal S., Hafeez I., Asokan N., Sadeghi A.-R., Tarkoma S., "IOT sentinel: Automated device-type identification for security enforcement in IoT", *Proceedings of the I37th International IEEE Conference on Distributed Computing Systems (ICDCS),* pp. 2177-2184, Atlanta, G.A., U.S.A., 2017,.
13. Kumar R., Zhang X., Wang W., Khan R. U., Kumar J., Sharif A., "A multimodal malware detection technique for Android IoT devices using various features", *IEEE Access,* Vol. 7, pp. 64411-64430, 2019.
14. Koroniotis N., Moustafa N., Sitnikova E., "Forensics and deep learning mechanisms for botnets in Internet of Things: A Survey of challenges and solutions", *IEEE Access,* Vol. 7, pp. 61764-61785, 2019.
15. Alasmary H., Khormali A., Anwar A., Park J., Choi J., Abusnaina A., *et al.*, "Analyzing and detecting emerging Internet of Things malware: a graph-based approach," *IEEE Internet of Things Journal,* Vol. 6, pp. 8977-8988, 2019.
16. Nguyen H.-T., Ngo Q.-D., Le V.-H., "A novel graph-based approach for IoT botnet detection," *International Journal of Information Security,* pp. 1-11, 2019.
17. Vinayakumar R., Alazab M., Soman K., Poornachandran P., and Venkatraman S., "Robust intelligent malware detection using deep learning", *IEEE Access,* Vol. 7, pp. 46717-46738, 2019.
18. Yin W., Zhou H., Wang M., Jin Z., and Xu J., "A dynamic malware detection mechanism based on deep learning", *International Journal of Computer Science and Network Security,* vol. 18, pp. 96-102, 2018.
19. Aman N., Saleem Y., Abbasi F. H., and Shahzad F., "A hybrid approach for malware family classification". In Batten L., Kim D., Zhang X., Li G. (Eds.): *Applications and Techniques in Information Security, Communications in Computer and Information Science*, Vol 719, pp. 169-180, Springer, Singapore, 2017.
20. David O. E., Netanyahu N. S., "Deepsign: Deep learning for automatic malware signature generation and classification", *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1-8. Killarney, Ireland, 12-17 July 2015,
21. Buczak A. L., Guven E., "A survey of data mining and machine learning methods for cyber security intrusion detection", *IEEE Communications Surveys and Tutorials,* Vol. 18, pp. 1153-1176, 2015.

22. Koroniotis N., Moustafa N., Sitnikova E., and Turnbull B., "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset", *Future Generation Computer Systems,* Vol. 100, pp. 779-796, 2019.
23. Hamza A., Gharakheili H. H., Benson T. A., and Sivaraman V., "Detecting volumetric attacks on lot devices via sdn-based monitoring of mud activity", Pr*oceedings of the 2019 ACM Symposium on SDN Research*, pp. 36-48, San Jose, C.A., U.S.A., 3-4 April 2019.
24. Azmoodeh A., Dehghantanha A., Choo K.-K. R., "Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning", *IEEE Transactions on Sustainable Computing,* Vol. 4, pp. 88-95, 2018.
25. Idika N., Mathur A.P., "A Survey of Malware Detection Techniques", Department of Computer Science, Purdue University, 2007.
26. Nguyen T. D., Marchal S., Miettinen M., Dang M. H., Asokan N., and Sadeghi A.-R., "*Dïot: A crowdsourced self-learning approach for detecting compromised iot devices*", *arXiv preprint arXiv:1804.07474,* 2018.
27. HaddadPajouh H., Dehghantanha A., Khayami R., Choo K.-K. R., "A deep Recurrent Neural Network based approach for Internet of Things malware threat hunting", *Future Generation Computer Systems,* Vol. 85, pp. 88-96, 2018.
28. Takase H., Kobayashi R., Kato M., and Ohmura R., "A prototype implementation and evaluation of the malware detection mechanism for IoT devices using the processor information", *International Journal of Information Security,* Vol. 19, pp. 71-81, 2020.
29. Gandotra E., Bansal D., and Sofat S., "Malware analysis and classification: A survey", *Journal of Information Security,* Vol. 2014, 2014.
30. Moser A., Kruegel C., Kirda E., "Limits of static analysis for malware detection", *Proceedings of the Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*, pp. 421-430, Miami Beach, F.L., U.S.A., 2007.
31. Aslan Ö. A,. Samet R., "A Comprehensive Review on Malware Detection Approaches", *IEEE Access,* Vol. 8, pp. 6249-6271, 2020.
32. Chen F., Deng P., Wan J., Zhang D., Vasilakos A. V., Rong X., "Data mining for the internet of things: literature review and challenges", *International Journal of Distributed Sensor Networks,* Vol. 11, 2015.
33. Chohan N. S. (2019). *Introduction to Artificial Neural Networks (ANN).* Available: https://towardsdatascience.com/introduction-to-artificial-neural-networks-ann-1aea15775ef9
34. Mustafa D. N. (2020). *ToN_IoT Datasets*. Available: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-ton-iot-Datasets/
35. Tavallaee M., Bagheri E., Lu W., and Ghorbani A. A., "A detailed analysis of the KDD CUP 99 data set", *Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1-6, Ottawa, Canada, 8-10 July 2009.
36. Moustafa N., Slay J., "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)", *Military Communications and information Systems Conference (MilCIS)*, pp. 1-6, Canberra, Australia, 1-12 November 2015.