Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606

Volume 04 Issue 01
2022

Research Article
https://doi.org/10.56979/401/2022/107

Digital Forensics Analysis of IoT Nodes using Machine Learning

M Zeeshan Arshad[1], Hameedur Rahman[2], Junaid Tariq[3], Adnan Riaz[4], Azhar Imran[4], Amanullah Yasin[4] and Imran Ihsan[4]

[1]Department of Cybersecurity, Faculty of Computing & AI, Air University, Islamabad, 44000, Pakistan.
[2]Department of Computer Games Development, Faculty of Computing & AI, Air University, Islamabad, 44000, Pakistan.
[3]Department of Computer Science, National University of Modern Languages, Rawalpindi, 43600, Pakistan.
[4]Department of Creative Technologies, Faculty of Computing & AI, Air University, Islamabad, 44000, Pakistan.
*Corresponding Author: Hameedur Rahman. Email: hameed.rahman@mail.au.edu.pk.

Received: September 21, 2022 **Accepted:** November 12, 2022 **Published:** December 29, 2022.

Abstract: With the versatility and exponential growth of IoT solutions, the probability of being attacked has increased. Resource constraint IoT devices raised a challenge for the security handler to track logs of different variety of attacks generated on them while performing the forensic analysis. Commonly forensic analysis is performed on the devices that calculate how much loss has occurred to the device due to the diversity of attacks. The main objective of this paper to develop a framework through which secueity can perfrom the forensic analysis on resource contraint IoT devices. In this paper, we have proposed a framework that intelligently performs forensic analysis and detects the different types of attacks performed on the endpoint (IoT device) using a node to node (N2N) framework. Furthermore, this proposed solution is a blend of different forensic tools and Machine learning techniques to identify different types of attacks. Using a third-party log server, the problem of evidence recovery from the endpoint under attack is addressed. To determine the nature and effect of the attack we have used the logs by using the security onion (forensic server). Additionally, this framework is equipped to automatically detect attacks by using the different machine learning algorithms. The efficiency of machine learning models is measured upon the values of (1) Accuracy, (2) Precision, (3) Recall, and (4) F-Measure. The results show that the decision tree algorithm stands out with the optimum performance compared to other ML models. Overall this framework can be used for the secuirty of IoT devices as well as the evidence collection from the IoT endpoint. For the validation of the proposed framework more detailed results and performance, analysis is presented in this paper.

Keywords: Cyber Security; Node-to-Node; Forensic analysis; Machine learning; Cyber attacks, Internet of things (IoT).

1. Introduction

The internet of things (IoT) is a system of interlinked devices that use common resources and "send and receive" data securely between them when an internet connection is established. In terms of straight-through processes, the IoT outclasses the conventional networks due to its expandable features and a much wider outlook. IoT has improved the standard of life of human beings by being pervasively utilized resulting in the invention of numerous innovations such as intelligent buildings, smart grid stations, wearable gadgets, smart appliances, and house automation.

It's a well-developed fact that IoT devices are improving exponentially continuously. However, security remains a challenge in general. The manufacturers are investing more time and resources in improving the functionalities and coming up with new features to attract a customer base rather than investing in addressing the security concerns. This lack of security interest has resulted in countless cyberattacks [1]. The other main causes of vulnerabilities in IoT devices are improper quality assurance testing, promptitude to release the product, and effective legislation inadequacy [2]. Denial of Service is one of the most common attacks on IoT networks. DDoS attacks are on a rise since 2018 and are still forecasted to increase more rapidly till 2023 as per Cisco's annual report. Figure- 1 shows the attacks recorded from 2018 to 2023 and the attacks forecasted in 2023 [3]. The Figure 1- shows a continuous increment in the trend of the attacks.

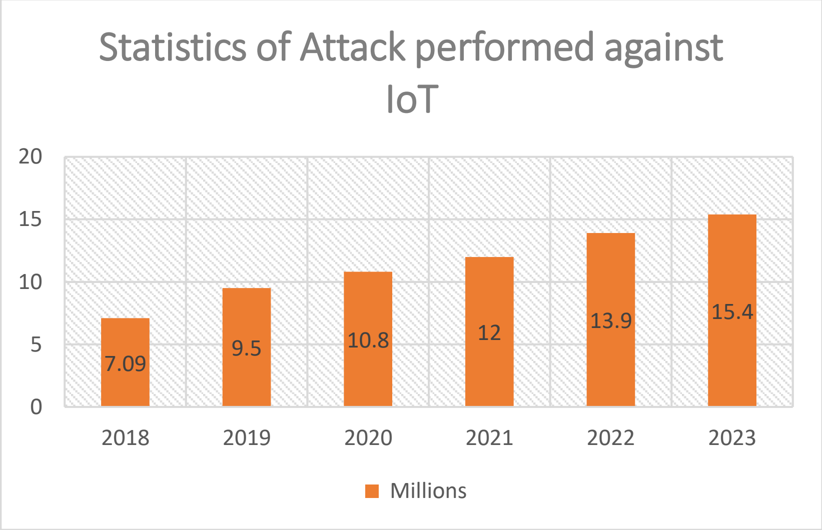

Figure 1. Statistics of Attack performed against IoT solutions (2018-2023).

The research team of PAN Unit Forty-Two has reported that up to 98 percent of such IoT devices are unencrypted causing multi-level attacks compromising confidential information [4]. The connection of these vulnerable devices to the network of the IoT system increases the attacker's threat surface. 1.5 billion such attacks have been indicated during the first half of 2021 by Kaspersky's research [5]. To further support many IoT devices being vulnerable, HP has reported up to 70 percent of such devices being insecure and penetrable [6]. Resource constraint IoT devices raised a challenge for the security analysts to track records of different variety of attacks generated on them while performing the forensic analysis along with the lack of acquisition of evidence being another major limitation [7]. This requires a framework to be established that can address the issue of detecting and storing evidence of such attacks. To apply more effective and improved forensic techniques, special approaches and tools are required so that the IoT environment becomes stronger and more secure [8].

For this, we put forward a framework of forensic analysis for the IoT devices that generates logs and send alerts regarding such attacks post detecting them automatically. Forensic analysis has a close relation to SIM as it refers to the thorough investigation of a particular crime post it had taken place to find the causes behind it such as the perpetrator, motives, and complex outcomes of the security incident. [9][10] However, Forensic analysis is almost the opposite of network auditing as the earlier is a post-incident analysis of security violation to document the course of action and the time of the breach while the latter is a pre-hand analysis of the flaws in a specific network [11]. In the proposed framework, the limitation of acquiring the data is addressed by using a 3rd party server for logging. The flow (packets) of IoT nodes is diverted to the said machine where it keeps files and generates alerts of malicious activity to be analyzed during forensic analysis. The track record of these logs is pulled up in an allotted forensic server and analyzed to get the information about the offenders and the attacks. The machine learning technique is used to automatically detect the attacks by providing the dataset of the logs.

There are four basic steps of forensic analysis in general, which are the acquisition of data, assessment, analysis, documentation, and report generation [12]. The major problem was faced while acquiring the data due to the IoT devices' inadequate processing power [13]. We introduced a logging server in our proposed framework that utilized snort to detect the attacks and keep a track record of malicious activity logs along with generating the alerts. In the assessment phase, a perusal of collected data is done to retrieve the relevant information. We used an IP table in the IoT device configuration to redirect the traffic. Pre-written logs were used by snort logging servers for attacks, and they were integrated into the detection engine. The detection engine filters the information by extracting the relevant data and discarding the rest of the data. Security Onion is used to providing the particulars of inspection such as the type of attack, details of the attacker, source and destination ports, and a lot of other useful information. In documentation and report generation, the conclusion of the analysis is provided. The knowledge obtained from the forensic procedure should be considered during future forensic operations [14].

This proposed framework incorporates both forensic tools and Machine Learning algorithms. By using the sniffing mode, Security onion, a forensic server, triggers new alerts for any malicious activity or traffic on the network. It also helps to formulate rule to detect attacks. These new rules can then be added into the suggested framework helping the ML to detect the attacks automatically. This then generates multiple reports describing the details of attack type and the frequency of an attack recommending actions to be taken for future reference. This provides a complete picture of the attacks along the trail of the attackers. We have used multiple forensic tools and machine learning algorithms for the proposed framework. The fundamental contribution of the article is enhancing conventional techniques of fetching and assessing traces of attacks from IoT nodes, which are beneficial for forensic analysis, to address the issue of low memory and low processing power of IoT devices' utilization of logging servers to keep the efficiency and security intact. One of the pivotal advantages of this framework is to compute, assess and analyze a huge set of data without affecting the performance or quality of the IoT devices [15]. For broader prospect any IoT empowered organization can use this method to generate the dataset of IoT attacks helping it to observe and analyze the malevolent activities performed [16]. This framework also keeps the track record of the cybercrimes, and the recorded information can be used as evidence against the perpetrator in the Legal Courts. Last but not the least Security analysts can utilize this combination of Machine learning and forensic tools framework to create effective systems to efficiently detect IoT attacks.

This paper is organized in the following manner. Section II consists of a literature review that gives an overview of various types of attacks on IoT systems, device and network-level forensic techniques, and intrusion detection systems for IoT. Section III contains the methodology on which our system is achieving the optimum results. Section IV explains the results achieved from both forensics and ML models applied, applying both in an integrated form produced the desired results. Section V included the conclusion of this research paper and discussed the future work related to this topic.

2. Literature Review

A comprehensive survey highlights the solutions and vulnerabilities related to privacy and routing in IoT solutions [17]. In other research [18] author discussed the DDoS attack performed on the IoT networks, as well as on MAC. In addition to that, he also discussed the intrusion detection system (IDS) for IoT networks. The author of the [19] article addressed the resource constraint problem in signature-based IDS by adapting the Suricata-based signature IDS, with the help of an independent Linux server. At the same time, the author did not manage to keep up-to-date signatures. Another researcher [20] further added that [19] in the signature-based matching techniques. In research [21] to minimize the operation of matching signatures is proposed by using table shift with pattern identification methods. The system used the repositories signatures by combining the snort as an IDS and ClamAV as an antivirus scanner.

Fabiola and HS. Venter [22] had advised digital forensic readiness structure which was operated under ISO/IEC 27043:2015. This structure consisted of 6 steps which consist of the packets capturing, gather data, PSO DL model, detect attack using ML, and evaluate the results. This paper instilled a simple and generic structure which was very useful to gather digital evidence by taking into consideration, the intricate features, processes, behaviors, and functionalities of the IoT devices. The researchers in [23] introduced a structure known as particle deep framework. This structure consisted of 5 steps flowchart. The data is collected in this structure post it is captured and directed by specific network capturing tools such as

Tcpdump, Ettercap, and Wireshark. Then PSO (particle swarm optimization) algorithm is put into use for the automation of deep learning dimensions. Lastly, DNN (deep neural network) is established based upon the PSO algorithm to identify and track down the IoT device's cyber-attacks. This model has an overall accuracy of 98 percent. Expert in this area [24] had come up with a novel approach to the identification of IoT devices under forensic investigation by introducing the term genes or DNA of devices. The DNA consists of two basic data i.e., the Universal device identification number and the buyer's details. Whenever an IoT device with DNA is used for any cyber-attack, it becomes very easy to track it down. Their model was tested on a Hybrid Forensic Server of IoT devices where each device was registered, and hence forensic analysis was easily applied.

Conventional forensic systems are considered inefficient due to limited hardware resources with the passage of time making them absolute, as we are heading towards the age of digital transformation. With the advancement of hardware and software attackers are using advanced methods to attack digital systems, critical infrastructure, and servers. Hence the forensic methods lack efficiency due to the connectivity of all devices which produces a relatively small amount of evidence. So, theirs is a need for time to pay significant attention to developing the latest, efficient, and automatic forensics analysis methods or techniques. Lots of tools are available for digital forensics according to the type of incident. The Forensic process is discussed concerning the tools and techniques that are required to fulfill the present-day needs. Also, the need for enhancements of the already existing frameworks. This research highlights the importance of forensic analysis in the newly emerging fields such as cloud-based systems, IoT Systems, and big data as well as with the already existing infrastructure as database servers, storage devices, OS, and mobile [25].

Another author of [26] studies various malware observation techniques along with their advantages and misuses. Sign-based detection includes the formation of a backend database of various signs that are isolated values for hostile files. Whenever malware is observed, the signature of this detection is differentiated from already saved signatures in the database. Attitude-based observation includes the utilization of various tools to get the attitude of being or any malware systems. Certain aspects are obtained and tackled with the help of machine learning techniques. Heuristic-based observations include the human experience and various machine learning techniques, and it can detect the zero-day attack, but it can’t trace modern complex malware. Model examination depends on techniques that include examining behaviors and clustering files along with the same behavior which we can label as malware. Artificial intelligence can be utilized for malware identification with the help of deep learning [27-29]. These depend on various steps such as aspect extraction, and identification of neural network layers, and the results are examined to identify malware. In cloud-based identification includes sending and saving files to the cloud and malware can be identified depending on behavior and stored signatures. There are also mobile device identification techniques in malware detection particularly various devices of android that utilized multiple aspects to give input to ML. IoT nodes are more vulnerable to attacks, as no proper security protocols are implemented due to resource constraints devices.

3. Methodology

To perform the forensic analysis of IoT nodes being attacked, the proposed framework consists of 4 modules, as shown in Figure- 2, Designed framework of IoT node to node, all the different components are connected with each other. IoT node can communicate bi-directionally, with all the other components through IoT hub and Gateway. First module is the attack’s traffic generation which consist of the Kali Linux OS with some exploiting tools are used to launch an attack on the IoT nodes for experimentation, which results in the generation of attacker's traffic. After the traffic being generated, this generated attacker’s traffic is passed to module called Traffic redirection and logs/alerts generation, this will redirect the traffic from the IoT node to the log server, this server is equipped to analyze the traffic, upon the matching of the rules written logs/alerts generated in the logging server. Re-generation of logs takes, which already captured in the logging server, vital information about the attacker and attack is gathered in this module, further these re-generated logs will be used as a dataset this module is named as forensic server. In the last module, Machine learning model applies different ML algorithms such as [30] RFC Classifier, [31] DT Classifier, [32] Naive Bayes Classifier, [33] LDA Classifier, [34] Modern language processing Classifier, and ensemble [35]. Voting Classifier are applied to the dataset generated in the forensic server module, which detects the attacks. These algorithm's/models are evaluated on different features.

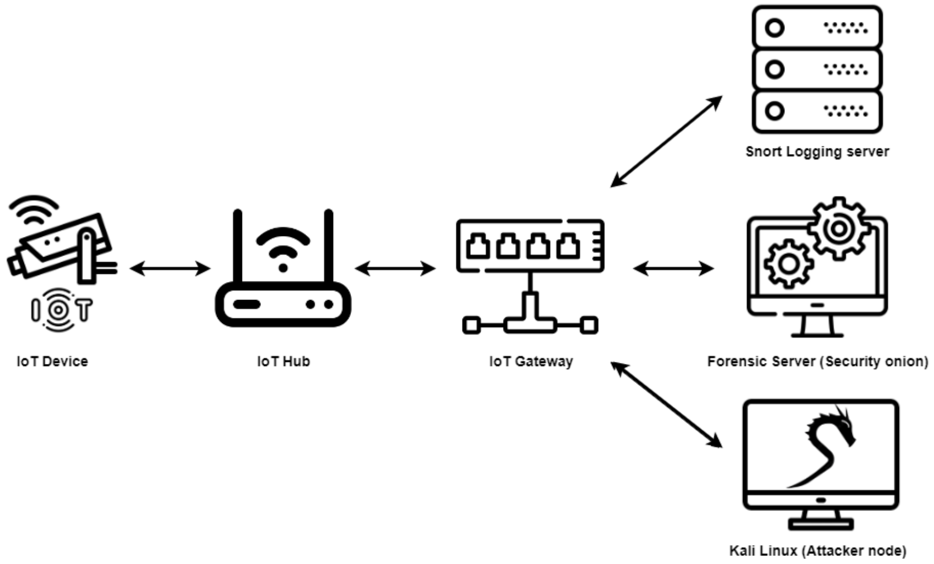

Figure 2. Designed framework of IoT node

For experimentation, we have used a few devices such as Raspberry Pi as IoT endpoint [36], for logging server snort [37], for forensic analysis server security onion [38], and kali Linux server is used to lunch attack on IoT endpoint. All the above-used nodes and devices relate to the same network. Figure- 3 shows the topology of the proposed framework that is proposed topology of the framework. Initially the attacker's traffic is generated to IoT device, incoming traffic is redirected to logging server. In this server snort rules detects the malicious traffic and logs generated, logs then converted to datasets and processed through ML plus security onion server. At the end combined results generated through ML and Forensic onion server.

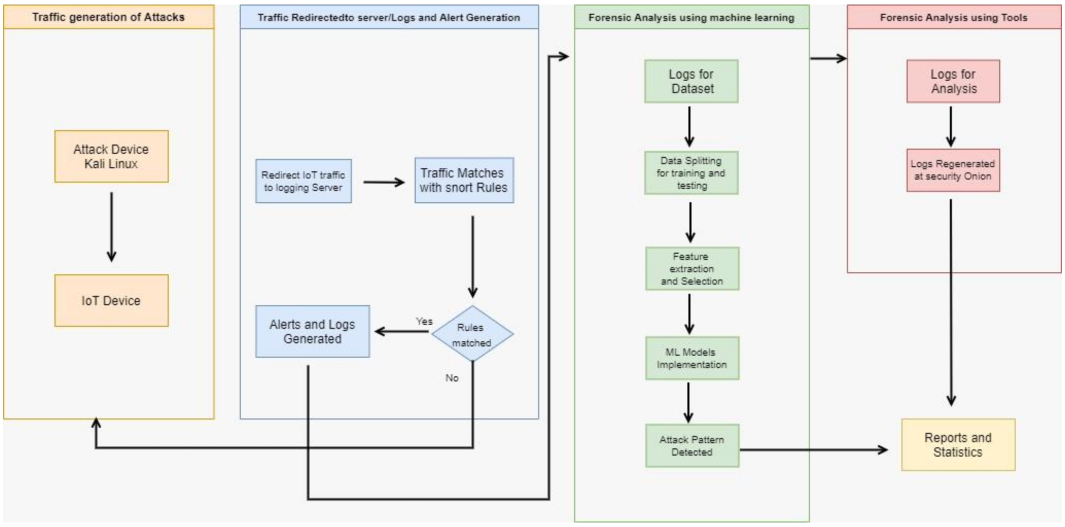
Traffic generation of Attacks
Attack Device Kali Linux
IoT Device
Traffic Redirectedto server/Logs and Alert Generation
Redirect IoT traffic to logging Server
Traffic Matches with snort Rules
Rules matched
Yes
No
Alerts and Logs Generated
Forensic Analysis using machine learning
Logs for Dataset
Data Splitting for training and testing
Feature extraction and Selection
ML Models Implementation
Attack Pattern Detected
Forensic Analysis using Tools
Logs for Analysis
Logs Regenerated at security Onion
Reports and Statistics

Figure 3. Proposed topology of the framework

3.1. Attack’s traffic generation

The proposed framework's first step is responsible to generate multiple attacks on the IoT node Raspberry Pi, as both node and the kali Linux are connected with the same network, and both shares the IP address of the same subnet this enables kali Linux to launch an attack. As IoT can connect with various types of devices such as cameras, RFID sensors, etc. We have built the IoT node on board with an open-source platform. Some of the Kali Linux tools such as Wireshark, NMAP, HPING3, Ettercap, and Metasploit are used in the generation of attacker’s traffic on the Raspberry Pi in the following sequence:

- NMAP is used for Port Scanning;
- Metasploit is used to lunch Brute-Force Attack;
- HPING3 is used for Syn Flooding to perform Denial of service DDoS;
- Ettercap is used for ARP spoofing to perform MITM (Man in The Middle).

3.2. Traffic redirection and logs/alerts generation

Proposed solution, traffic coming to the IoT node is transferred to the log machine to generate the logs. To handle the limitation of the IoT devices redirection is done based on the IPs of the connected devices. While implementation of the experiment’s nodes are connected in N-2-N manner so that it's feasible to share the data between all the devices. A third-party server is used for storing the logs which are done by a logging agent (WAZIH) [39]. Logging server is configured by rule defined and implemented by using snort tool. The ARM architecture of the IoT node does not support the redirection, to address this issue IP tables are used to redirect (IP-based) the traffic to the logging server. Snort-3.0.1 is used to analyze the redirected traffic; snort rules are configured in the configuration to detect the various types of attacks. Upon the successful match of the snort rule written in the logging server, a process is initiated that stores the logs in the log server, and a snort rule will be generate alerts. To perform forensic analysis with the ML techniques we cannot apply the ML technique on the pcap files generated by snort, to overcome this we have used the CIC flow meter open-source tool to convert pacp files to the CSV files dataset, now we can use these CSV files for the further processing.

3.3. Forensic Server

Once the attack is detected by the snort, logs are automatically stored in the logging server for analysis. Generated log files contain the user information such as source IP, destination IP, type of attack performed, etc. Security onion VM contains two network interfaces, among them, one is used for the functionality of the server, and the other is used for sniffing the packets to detect the malicious traffic. This whole mechanism of storing logs at the logging server addresses the problem of evidence acquisition. Security Onion has used a forensic server in our proposed framework, this comes with a variety of built-in tools such as Sguil which is a log analysis tool, and this tool is a GUI of snort. To acquire the information about the attacks we regenerate the logs, as we have already captured the logs using the Snort IDS.

3.4. Machine learning Model

To automate the attack detection of IoT nodes is done by applying the machine learning algorithm whereas the detection of attack using snort is a manual process, in which we run the IDS manually for various types of attacks. Automatically detection of attacks is achieved by the apply machine learning process, for which a labeled dataset in the form of logs (CSV files) is available. Then the labeled data is split into training and testing datasets shown in Figure 4, which is dataset generated through logging server is then processed through ML model. Labeled Dataset is spitted into training and testing datasets, in training feature extraction is performed and depending upon the features ML models are trained. In the testing part, the same process is repeated with the testing dataset, which validates the trained model. The next step is the feature extraction from the data. After this ML model is trained on the training data and tested with training and real-world data.

1. *Flow accumulation and labled data:* Logs generated are in the PCAP which cannot be used for the processing of the machine learning model, so we used a tool CIC flowmeter to convert the .PCAP log files into .CSV form. After extracting the features from the dataset, then this collected information is fed to

the ML model for training to identify threats to the IoT node. In the end, data is labeled based on its type and category to detect the legitimate and malicious behavior as shown in Table 1;

Table 1. ML Data Labeling.

Type	Category	Labels
Legitimate	Normal	0
Malicious	DDoS Syn-Flood	1
Malicious	Brute force	2
Malicious	MITM ARP Spoofing	3
Malicious	Ports Scan	4

2. *Pre-processing data:* The main aim of data preprocessing is to achieve good performance of the predictive ML models. Unnecessary fields, which do not contribute to classification have been removed from the dataset and the numeral features are scaled between zero to one. Additionally null or empty and outlier values are injected with appropriate values by using the statistical methods. Fields that have already been used for the labeling of the data must be removed because this will lead to the basis of the data and result in the degradation of performance;
3. *Splitting dataset into test dataset and training dataset:* Now split the dataset into a training dataset and test dataset, the ML model is trained on the trained using the training data set, and to verify the trained model accuracy testing dataset is used. This ratio in which splitting takes place is 7:3, training and testing respectively;
4. *Extraction and selection of Feature:* Correlation in the features of the dataset results in the degradation of the efficiency of the machine learning algorithms. We used k-best [40], Correlation coefficient, backward elimination, information gain, and feature importance to select features. We have selected the k-best for feature extraction by setting the k = 10 to get the best accurate results;
5. *Training of ML model*: Firstly, ML methods simply get inputs from already extracted features. Various methods are used to implement ML methods such as best installation. The performance of ML mostly depends on pre-trained data which are trained using various techniques. The working of ML depends on a few steps such as (1) first giving the input raw data to ML to get the specific required features, (2) then these features forward towards a trained model for label predictions, as before we simply did with our data. As the result, an attack that occurs can easily be detected from these predicted labels. The architecture of the above-mentioned procedure is shown below Figure-, In this article, we utilized various assessment techniques to get out the performance of our models. For this purpose, we utilized the various aspects of the confusion matrix such as (1) accuracy, (2) precision, (3) recall, and the last one is (4) F1-score. If we utilized the trained dataset, then performance will not be considered of error. The technique used to calibrate the model is cross-validation.
6. *Reporting:* The main area of our interest is to develop node-2-node interaction depending on the IoT machine. The procedure of our model is to give specific IP to IoT, and the remaining machines are on restricted architecture. The purpose of developing such a domain which focus the forensic based on IoT architecture utilizing node-2-node connections. In this scenario, Various attacks are performed on IoT machines [41-43]. The function of our proposed architecture is to alter the coming traffic for IoT towards snort. The logging server placed there is build the logs and forwards them to the server for the forensic procedure, where packets are revived and examined. To evaluate this proposed solution pre-trained dataset is utilized for ML. identification of attack enhancement is more effective and comes within time with the combined working of ML and forensic models shown in Figure- 5. After these all implantation, the various ledger is developed with complete descriptions related to the attack and how many times it occurs and at the end, suggested action that needs to be implemented. The report helps us to highlight and draw the whole attack script and make it possible to reach out to attackers.

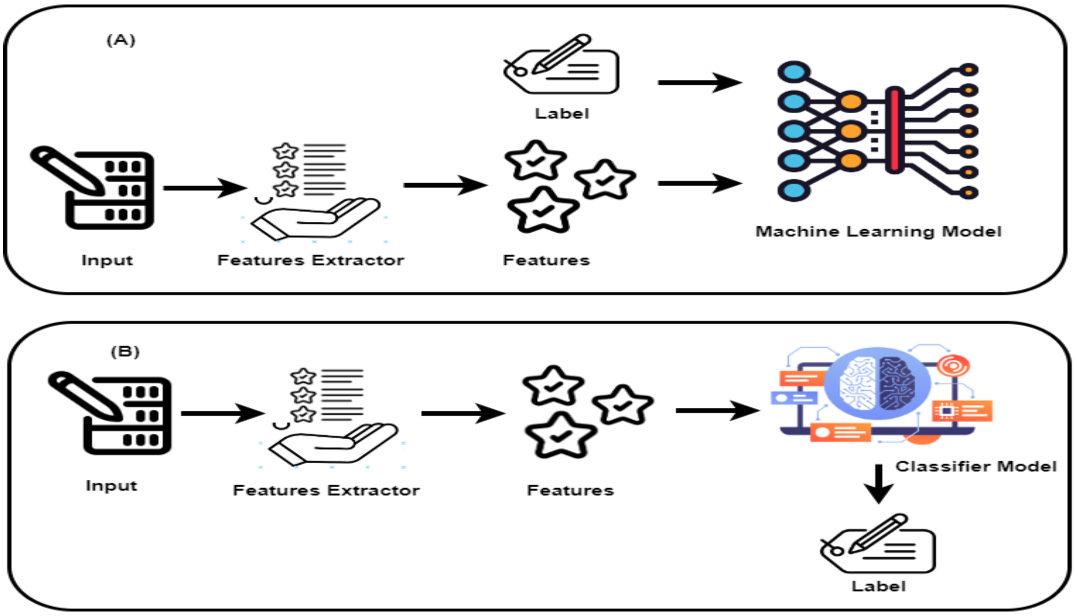

Figure 4: (a) Training ML Model, (b) Testing of the trained model.

4. Discussion

The proposed solution contains the forensic analysis module which is responsible for the automatic attacks detection that occurred on the IoT node. Considering the resource limitations of the IoT nodes, an embedded forensic analysis mechanism cannot be used. A logging server is used to carter this issue. In this article, various types of attacks are performed on IoT nodes. IP tables are to redirect the incoming packets to the logging server. Snort rules written at the logging server match the attached performed. The main functionality of the logging server is to produce alerts/logs and store the generated logs for further process of forensic analysis. Both ML models and forensic onion server is used for forensic analysis.

4.1. Forensic Server

First, various types of attacks are performed on the IoT endpoint mentioned in the architecture diagram. Kali Linux utilities are used to launch the attack, then the malicious traffic is identified by the snort on the logging server. Identified malicious traffic with the help of snort rules is saved to a log file in the extension of pcap with a different name each time. Then these log files are transformed into a dataset using the CIC flow tool to CSV extension for the further processing in the ML models. For the forensic analysis, we have used the security onion which is a Linux-based Operation system. This OS is equipped with security-related tools whose purpose is to provide log management, features of IDS, and monitoring. Log interpreter tools helped to visualize the log files. Even with going down deep in the logs, these tools help us to take overlook the different types of attacks performed differentiating with the color scheme. Squert is used for the analysis logs of DDoS attacks, it shows numerically how many times an SSH request is initiated by the attacker and also shows the attacker is not intended to establish the connection, but only flood the endpoint. Generated logs also show the port scanning for open ports, information about OS, Mac address, and SSH key are tried to be obtained by the attacker.

4.2. Machine learning model

ML models are used for the automatic detection of attacks against the IoT nodes. For this purpose, we have used various ML techniques, this is applied to the dataset that is transformed from the logs that are already generated from the logging server. To achieve the optimum results, we have applied numerous techniques for the feature extraction from the dataset. We used k-best, Correlation coefficient, backward elimination, information gain, and feature importance to select features. The splitting of the dataset is trained to test a ratio is 7:3. To validate different ML models we have compared on the basis: of (1) Accuracy, (2) Precision, (3) Recall, and (4) F-Measure. The table shows the numerical values of the above-mentioned parameter.

As mentioned above various feature selection techniques are applied, before applying these we have cleaned the data by discarding the fields with a minimum contribution to the required output. Splitting is done 70 for the training and 30 for the testing, this same splitting is used to train the multiple ML classifiers, among these DT (decision tree) turned out to be the best fit for our dataset. DT stands out from other ML algorithms with the highest accuracy of 97.29 percent. Comparison of the accuracies of the used ML model as drawn in the Figure- 6 explains the architecture of forensic and logging server. The incoming traffic to the IoT device is being re-directed to the snort logging server with the snort rules configured on it, makes a complete detection Engine. Upon the basis of rules defined it categorize it malicious or normal traffic, logs and alerts generated in case of malicious traffic and passed on to the forensic server which results in the generation of regenerated log files for analysis.

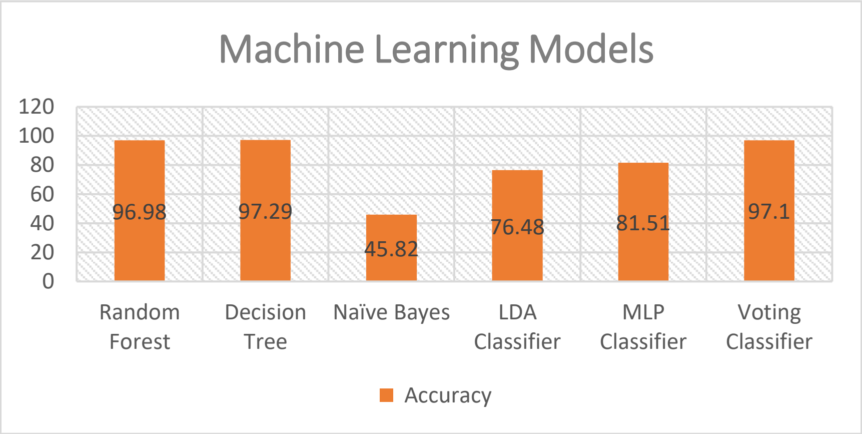

Figure 5. Analysis of Accuracy comparison.

To classify the between the legitimate and malicious packets various type of classification ML models have been used, the accuracy of 88 percent had achieved by the proposed models. Contrary to this our proposed methodology, in the combination of forensic server logs generated along with forensic server tools used as a dataset with the ML techniques gives the output of 97.29 percent accuracy. These machine learning methods has already achieved state-of-the-art results for various problems [44-50.

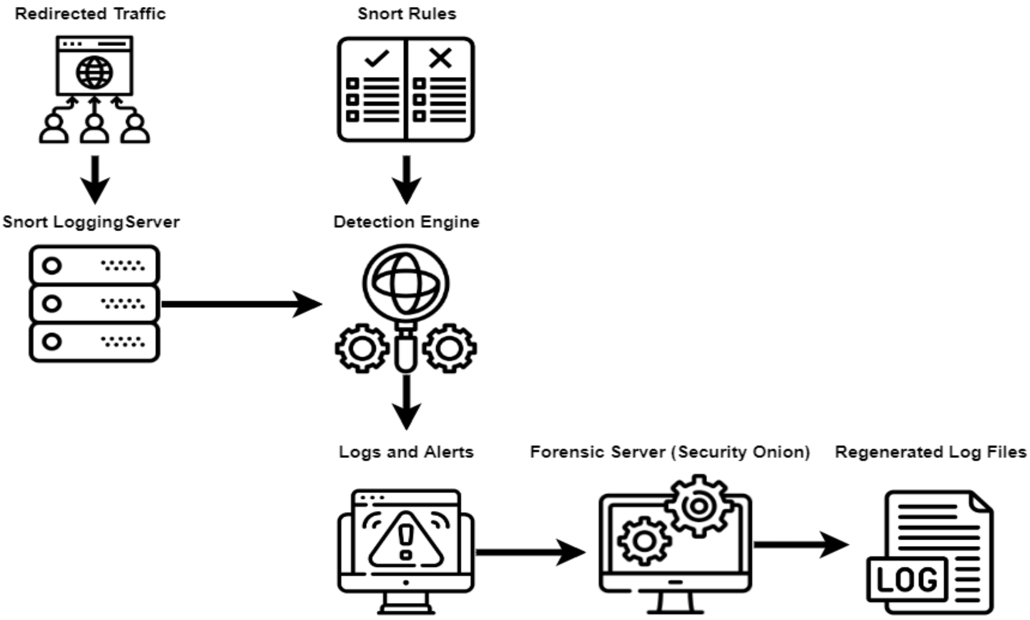

Figure 6. Architecture of forensic and logging server.

5. Conclusions

Forensic investigation is the thorough inspection of the crime post it had committed to find the causes behind it. Our proposed framework addresses the issues of low memory and low storage of IoT devices effectively. The proposed framework operates in a straight-through process environment making it more time-efficient and dependable. Network traffic is transferred to logging servers with continuous communication among devices. Rules are set at the logging server to draw a comparison with network traffic and then analyze it accordingly. The logs of malevolent traffic are kept safe and can be pulled up through various methods at the forensic server. Furthermore, a dataset of these attacks is also generated and subsequently, this dataset is used to detect attacks by utilizing several machine learning schemes. We had the highest accuracy for the decision tree structure which was at 97.29 percent. We had installed a Pi camera in our system to test our proposed framework in a real-time scenario. The overall efficiency of the machine learning structure dropped slightly post the installation of the Pi camera with the decision tree structure's accuracy still the highest at 96.01 percent. Multiple reports describing the information about the attack types, frequency of attacks, and the suggested actions to be taken were then generated. This detail will make the tracing of attackers possible by drawing a complete picture of the attack footprints.

The horizon of this article will be expanded by including high rate of attack with segregation of classifications and sub-classifications. Used data set comprises common IoT device-based attacks which makes it a limited set of data. The features of this model can be upgraded by applying more advanced techniques. The dataset of daily used IoT devices can also be added to improve ML-based forensic analysis' footprint. In future to generalize the ML trained model for various types of IoT devices and variety of attacks, various type of IoT devices such heart-rate monitoring, glucose monitoring, connected inhalers and ingestible sensors can be added in the framework to generate dataset for forensic analysis.

References

1. Sikder, A., Petracca, G., Aksu, H., Jaeger, T. and Uluagac, A., 2021. A Survey on Sensor-Based Threats and Attacks to Smart Devices and Applications. *IEEE Communications Surveys &amp; Tutorials*, 23(2), pp.1125-1159.
2. Tawalbeh, L., Muheidat, F., Tawalbeh, M. and Quwaider, M., 2020. IoT Privacy and Security: Challenges and Solutions. *Applied Sciences*, 10(12), p.4102.
3. Hussain, F., Abbas, S. G., Husnain, M., Fayyaz, U. U., Shahzad, F., & Shah, G. A. (2020, November). IoT DoS and DDoS attack detection using ResNet. In *2020 IEEE 23rd International Multitopic Conference (INMIC)* (pp. 1-6). IEEE.
4. More Than Half of IoT Devices Vulnerable to Severe Attacks. (2022). Retrieved 29 September 2022, from https://threatpost.com/half-iot-devices-vulnerable-severe-attacks/153609/.
5. O'Donnell, A.L. &amp; O'Donnell, L., More than half of IOT devices vulnerable to severe attacks. Threatpost English Global threatpostcom. Available at: https://threatpost.com/half-iot-devices-vulnerable-severe-attacks/153609/.
6. Asif, S., Ambreen, M., Muhammad, Z., ur Rahman, H., & Iqbal, S. Z. (2022). Cloud Computing in Healthcare-Investigation of Threats, Vulnerabilities, Future Challenges and Counter Measure. LC International Journal of STEM (ISSN: 2708-7123), 3(1), 63-74.
7. Rughani, P. H. (2017). IoT evidence acquisition—Issues and challenges. *Advances in Computational Sciences and Technology*, *10*(5), 1285-1293.
8. Karabiyik, U., & Akkaya, K. (2019). Digital forensics for IoT and WSNS. In *Mission-oriented sensor networks and systems: Art and science* (pp. 171-207). Springer, Cham.
9. Anon, Forensic analysis. Forensic Analysis - an overview | ScienceDirect Topics. Available at: https://www.sciencedirect.com/topics/chemistry/forensic-analysis.
10. Lord, N. (2018). What is Security Incident Management? The Cybersecurity Incident Management Process, Examples, Best Practices, and More. *Digital Guardian's Data Insider*.
11. Rahman, H., Arshad, H., Mahmud, R., & Mahayuddin, Z. R. (2017, October). A framework for breast cancer visualization using augmented reality x-ray vision technique in mobile technology. In AIP Conference Proceedings (Vol. 1891, No. 1, p. 020116). AIP Publishing LLC.
12. Haider, S. K., Jiang, A., Almogren, A., Rehman, A. U., Ahmed, A., Khan, W. U., & Hamam, H. (2021). Energy efficient UAV flight path model for cluster head selection in next-generation wireless sensor networks. *Sensors*, *21*(24), 8445.
13. Horsman, G. (2022). An "order of data acquisition" for digital forensic investigations. *Journal of Forensic Sciences*, *67*(3), 1215-1220.
14. Ghabban, F. M., Alfadli, I. M., Ameerbakhsh, O., AbuAli, A. N., Al-Dhaqm, A., & Al-Khasawneh, M. A. (2021, June). Comparative analysis of network forensic tools and network forensics processes. In *2021 2nd International Conference on Smart Computing and Electronic Enterprise (ICSCEE)* (pp. 78-83). IEEE.
15. Zia, Z. U. R., Rahman, H. U., Malik, M. H., & Jahngir, A. (2020). Technical Challenges in Achieving Ultra-Reliable & Low Latency Communication in 5G Cellular-V2X Systems. LC International Journal of STEM (ISSN: 2708-7123), 1(3), 89-95.
16. Abbas, M., Arshad, M., & Rahman, H. (2020). Detection of Breast Cancer Using Neural Networks. LC International Journal of STEM (ISSN: 2708-7123), 1(3), 75-88.
17. Hameed, S., Khan, F. I., & Hameed, B. (2019). Understanding security requirements and challenges in Internet of Things (IoT): A review. *Journal of Computer Networks and Communications*, *2019*.
18. Arıs, A., Oktug, S. F., & Voigt, T. (2018). Security of internet of things for a reliable internet of services.
19. Kasinathan, P., Pastrone, C., Spirito, M. A., & Vinkovits, M. (2013, October). Denial-of-Service detection in 6LoWPAN based Internet of Things. In *2013 IEEE 9th international conference on wireless and mobile computing, networking and communications (WiMob)* (pp. 600-607). IEEE.
20. Kasinathan, P., Costamagna, G., Khaleel, H., Pastrone, C., & Spirito, M. A. (2013, November). An IDS framework for internet of things empowered by 6LoWPAN. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security* (pp. 1337-1340).
21. Oh, D., Kim, D., & Ro, W. W. (2014). A malicious pattern detection engine for embedded security systems in the Internet of Things. *Sensors*, *14*(12), 24188-24211.
22. Fagbola, F. I., & Venter, H. S. (2022). Smart digital forensic readiness model for shadow IoT devices. *Applied Sciences*, *12*(2), 730.
23. Koroniotis, N., Moustafa, N., & Sitnikova, E. (2020). A new network forensic framework based on deep learning for Internet of Things networks: A particle deep framework. *Future Generation Computer Systems*, *110*, 91-106.
24. Scheidt, N., & Adda, M. (2020, August). Identification of iot devices for forensic investigation. In *2020 IEEE 10th International Conference on Intelligent Systems (IS)* (pp. 165-170). IEEE.
25. Patil, A., Banerjee, S., Jadhav, D., & Borkar, G. (2022). Roadmap of Digital Forensics Investigation Process with Discovery of Tools. *Cyber Security and Digital Forensics*, 241-269.
26. Aslan, Ö. A., & Samet, R. (2020). A comprehensive review on malware detection approaches. *IEEE Access*, *8*, 6249-6271.
27. Wahab, M. A. A., Surin, E. S. M., Nayan, N. M., & Rahman, H. (2021). MAPPING DEFORESTATION IN PERMANENT FOREST RESERVE OF PENINSULAR MALAYSIA WITH MULTI-TEMPORAL SAR IMAGERY AND U-NET BASED SEMANTIC SEGMENTATION. Malaysian Journal of Computer Science, 15-34.

28. Abid, I., Almakdi, S., Rahman, H., Almulihi, A., Alqahtani, A., Rajab, K., ... & Shaikh, A. (2022). A Convolutional Neural Network for Skin Lesion Segmentation Using Double U-Net Architecture. INTELLIGENT AUTOMATION AND SOFT COMPUTING, 33(3), 1407-1421.
29. Rahman, H., Bukht, T. F. N., Imran, A., Tariq, J., Tu, S., & Alzahrani, A. (2022). A Deep Learning Approach for Liver and Tumor Segmentation in CT Images Using ResUNet. Bioengineering, 9(8), 368.
30. Pal, M. (2005). Random forest classifier for remote sensing classification. *International journal of remote sensing*, *26*(1), 217-222.
31. Jagannathan, G., Pillaipakkamnatt, K., & Wright, R. N. (2009, December). A practical differentially private random decision tree classifier. In *2009 IEEE International Conference on Data Mining Workshops* (pp. 114-121). IEEE.
32. Feng, X., Li, S., Yuan, C., Zeng, P., & Sun, Y. (2018). Prediction of slope stability using naive Bayes classifier. *KSCE Journal of Civil Engineering*, *22*(3), 941-950.
33. Balakrishnama, S., & Ganapathiraju, A. (1998). Linear discriminant analysis-a brief tutorial. *Institute for Signal and information Processing*, *18*(1998), 1-8.
34. Windeatt, T. (2006). Accuracy/diversity and ensemble MLP classifier design. *IEEE Transactions on Neural Networks*, *17*(5), 1194-1211.
35. Ruta, D., & Gabrys, B. (2005). Classifier selection for majority voting. *Information fusion*, *6*(1), 63-81.
36. Pajankar, A. (2021). *Practical Linux with Raspberry Pi OS*. Apress.
37. Krishna, G. S., Kiran, T. S. R., & Srisaila, A. (2021). Testing performance of RaspberryPi as IDS using SNORT. *Materials Today: Proceedings*.
38. Heenan, R., & Moradpoor, N. (2016, May). Introduction to security onion. In *The First Post Graduate Cyber Security Symposium*.
39. GitHub. 2022. *GitHub - ahlashkari/CICFlowMeter: CICFlowmeter-V4.0 (formerly known as ISCXFlowMeter) is an Ethernet traffic Bi-flow generator and analyzer for anomaly detection that has been used in many Cybersecurity datsets such as Android Adware-General Malware dataset (CICAAGM2017), IPS/IDS dataset (CICIDS2017), Android Malware dataset (CICAndMal2017) and Distributed Denial of Service (CICDDoS2019).* [online] Available at: <https://github.com/ahlashkari/CICFlowMeter> [Accessed 29 September 2022].
40. Rahman, H., Arshad, H., Mahmud, R., Mahayuddin, Z. R., & Obeidy, W. K. (2017). A framework to visualize 3d breast tumor using x-ray vision technique in mobile augmented reality. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9(2-11), 145-149*).*
41. Salleh, S., Mahmud, R., Rahman, H., & Yasiran, S. S. (2017). Speed up Robust Features (SURF) with Principal Component Analysis-Support Vector Machine (PCA-SVM) for benign and malignant classifications. Journal of Fundamental and Applied Sciences, 9(5S), 624-643.
42. Obeidy, W. K., Arshad, H., Yee Tan, S., & Rahman, H. (2015, November). Developmental analysis of a markerless hybrid tracking technique for mobile augmented reality systems. In International Visual Informatics Conference (pp. 99-110). Springer, Cham.
43. Tariq, J., Alfalou, A., Ijaz, A., Ali, H., Ashraf, I., Rahman, H., ... & Rehman, S. (2022). Fast intra mode selection in HEVC using statistical model. Computers, Materials and Continua, 70(2), 3903-3918.
44. Mahmood, T., Li, J., Pei, Y., Akhtar, F., Imran, A., & Yaqub, M. An automatic detection and localization of mammographic microcalcifications ROI with multi-scale features using the radiomics analysis approach. Cancers, 2021, 13(23), 5916.
45. Imran, A., Nasir, A., Bilal, M., Sun, G., Alzahrani, A., & Almuhaimeed, A. Skin Cancer detection using Combined Decision of Deep Learners. IEEE Access, 2022.
46. Imran, A., Li, J., Pei, Y., Akhtar, F., Mahmood, T., & Zhang, L. Fundus image-based cataract classification using a hybrid convolutional and recurrent neural network. The Visual Computer, 2021, 37(8), 2407-2417.
47. Imran, A., Li, J., Pei, Y., Akhtar, F., Yang, J. J., & Dang, Y. Automated identification of cataract severity using retinal fundus images. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 2020, 8(6), 691-698.
48. Imran, A., Li, J., Pei, Y., Akhtar, F., Yang, J. J., & Wang, Q. Cataract detection and grading with retinal images using SOM-RBF neural network. In 2019 IEEE Symposium Series on Computational Intelligence (SSCI), 2019, 2626-2632. IEEE.
49. Imran, A., Li, J., Pei, Y., Mokbal, F. M., Yang, J. J., & Wang, Q. Enhanced intelligence using collective data augmentation for CNN based cataract detection. In International Conference on Frontier Computing, 2019, 148-160. Springer, Singapore.
50. Bilal, A., Sun, G., Mazhar, S., Imran, A., & Latif, J. A Transfer Learning and U-Net-based automatic detection of diabetic retinopathy from fundus images. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 2022, 1-12.