Machine Learning Based Malware Analysis in Digital Forensic with IoT Devices

Sreenidhi Ganachari(✉), Pramodini Nandigam, Anchal Daga, Sachi Nandan Mohanty, and S. V. Sudha

School of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh, India
sreenidhi.ganachari5@gmail.com

Abstract. The use of IoT (Internet of Things) devices such as echo devices, smart locks, hue lights amongst a few, in our daily lives, has increased widely in this era of digitalization. People are gradually becoming dependent on these devices for their work or to store confidential data. This has also led to the concerns of security that arise with the use of these IoT devices. IoT devices are prone to malware attacks because of their dependency on the internet, technical complexity and integration of both hardware and software technology. The use of vulnerabilities in these devices by the cyber criminals is becoming extravagant. Also, the identification and categorization of IoT malware by cybersecurity analysts is further complicated by the diversity of IoT malware and the heterogeneity of IoT platforms. The aim of this paper is to analyze the malwares that are affecting the IoT devices and propose machine learning methodologies to identify these malwares based on various parameters. This paper focused mainly on malwares such as Mirai, Torii, Mushtik and Trojan that have been rampant in IoT devices these days. The models were trained based on algorithms such as SVM, Decision Tree, Naive Bayes, CNN, XG Boosting Classifier and Gradient Boosting Regression. The XG Boosting Classifier model has provided the highest accuracy of 97.4% amongst all other models. Thus, for the dataset used, XG Boosting Classifier is the best classifier that can be used to detect malware traffic in IoT devices.

Keywords: IoT devices · Malware · XGBoost Classifier · Forensic · Machine Learning · Cybersecurity

1 Introduction

IoT promotes integration between the physical world and computer communication networks. Applications (apps) involving infrastructure management and environmental monitoring make privacy and security measures vital for future IoT systems. [4] The Internet of Things (IoT) is viewed as a technological and commercial wave in the worldwide information economy. It is an intelligent network that links all gadgets to the Internet so they may communicate and exchange information using information-sensing devices, in line with established standards. It succeeds in detecting, locating, tracking,

S. Nandan Mohanty et al. (Eds.): ICISML 2022, LNICST 470, pp. 169–183, 2023.
https://doi.org/10.1007/978-3-031-35078-8_15

monitoring, and managing objects intelligently [3]. Figure 1 shows the different uses of IoT in various industries.

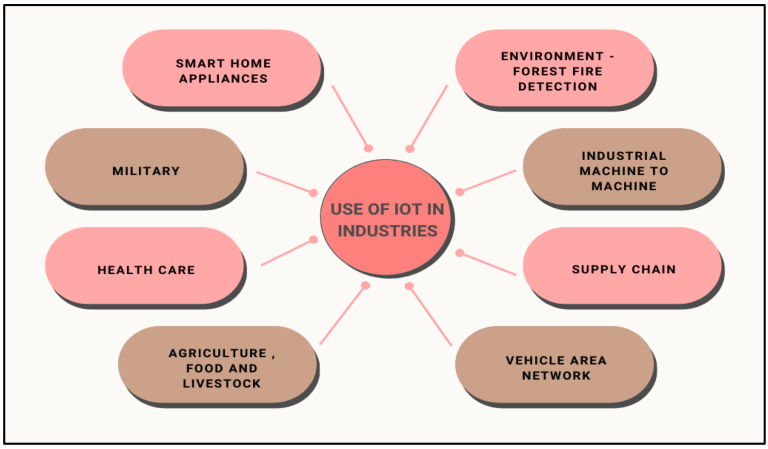

Fig. 1. Use of IoT in industries

Deployment of IoT devices has surged in the recent past and it continues to show staggering increase. These smart devices have facilitated growth for every sector and household. This sudden upsurge in technology however, also makes these devices susceptible to multiple malware attacks because of their straightforward implementation, default password settings, and hard-to-patch features [5]. As shown in Fig. 2, IoT devices have been steadily increasing over the years and will continue to grow further due to the heavy dependence on its applications. The information stored on these devices might be easily accessible to the attacker which then leads to dangerous use of sensitive personal information by the attacker. For example, A cybercriminal gaining easy access to a home security device can very well break in, which is a major security risk for the residents and the property itself. With minor flaws in technology, there might be a roadblock in the advancement of IoT devices. There are many factors that determine malware attacks such as connection of multiple IoT devices to one server, weak firewalls, unencrypted data, anti-virus not used and setting weak passwords, amongst others. It is therefore necessary to strengthen the security and privacy of these devices. The existing detection methods are not inefficient to identify the newer, more prevalent malwares.

In this paper, we aim to work on malware analysis i.e., a method or research that identifies the source, functioning, and potential consequences of a malware specimen. Depending on the sort of strategy used, IoT malware detection methods can be divided into two primary categories: dynamic analysis and static analysis. The dynamic technique entails keeping an eye on executables while they are running and spotting strange activities. Numerous resources are required to monitor running processes, and malware could occasionally invade real situations. By analyzing and identifying harmful files without running them, the static technique is used. A significant benefit is its capacity to observe hardware architecture. With the development of Artificial Intelligence, researchers are able to develop advanced models based on Machine Learning and Deep

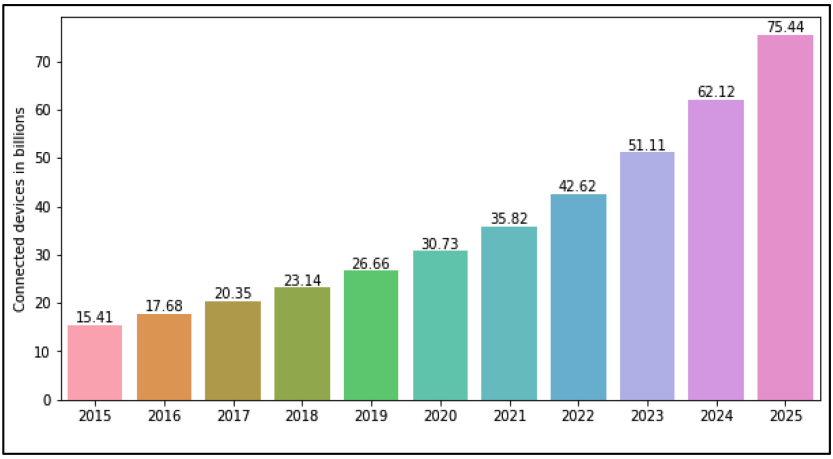

Fig. 2. Exponential increase of the use of IoT devices over years

Learning Algorithms. Researchers train multiple models to look for the best model that suits that particular dataset. There are multiple malware patches that can be detected with the use of these algorithms. This work uses multiple Machine Learning Algorithms such as Decision Tree classifier, Naive Bayes classifier, SVM classifier, CNN, Gradient Boosting and XGBoost. The various factors that determine the presence of malware in the captures are also discussed in this paper.

2 Related Work

In [6] the researchers present a CNN to extract the appropriate features from the dataset and make use of the Long Short-Term Memory (LSTM) model to classify the data. Although the researchers have built a model that outperforms many DL models of recent times, they have used a small amount of data samples. The model is also trained using only the LSTM model which can be made even more efficient both in terms of the dataset and algorithm used.

In [13] the authors have used CNNGRU machine learning algorithm to work on six datasets to obtain the accuracy. It includes binary classification of data which is efficient with an accuracy of 99.92. But here the authors have tested the model on a small dataset and the classifier does not classify the different IoT malwares due to the binary model design. This is a shortcoming in the paper because without knowing the malware type specifically we cannot mitigate it.

SVM suggested in [27] to be used to identify attacks that affect the volume of IoT network traffic, which are frequent in DoS. On simulated datasets the performance of several SVM kernels was examined. Despite the fact that the Linear kernel achieved 98.03% accuracy after a brief training period, this method is unable to identify attacks that neither rise nor decrease traffic intensity.

In [16] The most recent network traffic datasets were used to discover abnormalities using a combination of DNN and LSTM, followed by a meta-classifier. If more complex

datasets are available, the implementation technique can be expanded to run tests on them. Future network data processing will be accelerated and made more scalable through the use of cutting-edge computing techniques like Apache Spark. The method must also be verified for use in multi-class problem solving.

In [22] evaluated the effectiveness of ML for attack detection using a variety of machine learning models. They used a variety of ML classifiers, including Decision Tree, Logistic Regression, Random Forest Naive Bayes, and Support Vector Machine. According to test results, DT and RF had the highest accuracy detection rate of 1.00%, whereas LR, SVM, and NB classifiers only managed to reach 0.76%, 0.74%, and 58% accuracy, respectively.

In [7], the researchers examined the usability of ML techniques to anomaly detection with a focus on the security aspect of Internet of Things networks. The study made use of 14 features from the IoT-23 dataset. In terms of outcomes, Random Forest, another algorithm, has a weighted average precision of 100%, AdaBoost, another method, has a precision of 86%, Support Vector Machine, a precision of 60%, and Naive Bayes, a weighted average precision of 76%.

Researchers in [26] trained multiple models for anomaly detection resembling an industrial operation. Random Forest, had an accuracy of 97.44%, had the best overall performance. The only algorithm to achieve a False Positive Rate (FPR) of 0.00, or zero false alarms, was SVM.

On the IoT-23 dataset, the researchers assessed the efficacy of machine learning classifiers based cyber security approaches in [25]. For IoT cyber security in 2021, they employed RF, SVM, and KNN algorithms using seven features from the IoT-23 dataset. Their malware detection results show that SVM achieved 83.52%, KNN achieved 89.80%, and RF obtained 92.27% accuracy.

In [24] The authors have used two datasets for classifying IOT botnets and limited their work to 4 malware botnets while considering 12 features by preprocessing the data with KNN algorithm & generating new samples using CTGAN. Higher accuracies were obtained using their methodology, but considering a label ‘Malicious’ contains collective data of all kind of malicious traffic which leads to failure in detecting the peculiar malicious content specifically.

3 Proposed Methodology

3.1 Data Description

The dataset that this paper incorporated as part of the study, was initially created as part of the Avast AIC laboratory with the funding of Avast Software. IoT-23 is a dataset of Internet of Things (IoT) devices network traffic. In IoT devices, it has captured 20 malware executions and 3 benign IoT device traffic grabs. With pictures from 2018 to 2019, it was initially released in January 2020. The Stratosphere Laboratory, AIC group, FEL, CTU University, Czech Republic, is where this Internet of Things network traffic was recorded. Its objective is to provide researchers working on machine learning algorithms with a sizable dataset of actual, labeled IoT malware infections and IoT innocuous traffic. Avast Software, Prague, provided funding for this dataset and related research.

The IoT-23 dataset is made up of twenty-three different IoT network traffic scenarios (referred to as captures). Twenty network captures (pcap files) from infected IoT devices are used for these scenarios, and three network captures of network traffic from actual IoT devices were included. Each network capture provided the identity of the malware sample that was used to perform the scenario. In each malicious scenario, Raspberry Pi was used to execute a specific malware sample by using several protocols.

Three separate IoT devices were used to collect the network traffic used in the benign scenarios: an Amazon Echo home intelligent personal assistant, a Somfy smart door lock, and a Philips HUE smart LED bulb. It is crucial to note that these three Internet of Things devices are genuine hardware, not simulations.

As a result, one can record and examine actual network behavior. Like any other genuine IoT device, both malicious and benign scenarios function in a controlled network environment with unrestricted internet access. This dataset's purpose is to make two different datasets available to the community: the first comprises only benign IoT traffic, while the second exclusively contains malicious network traffic. There are two new columns for network behavior description labels for both good and bad traffic flows. This dataset additionally includes labels that indicate the relationship between flows connected to dangerous or potentially malicious activity in order to give network malware researchers and analysts more in-depth information. The Stratosphere laboratory developed these labels after analyzing malware captures. Brief descriptions of the labels used for manual network analysis-based identification of malicious flows –

a) Attack – This label denotes that a host-to-host attack from the infected device occurred (attack was launched from the infected device to another host). Any flow that attempts to exploit a service that is susceptible/weak by analyzing its payload and behavior is labeled as an attack.
b) Benign – This label denotes that they're no indications of not so harmful malicious activities that were discovered in the flow or connections.
c) Command & Control – [9] This label indicates that the infected device was connecting to a CC server. This activity was found during the investigation of the network malware capture because connections to the suspicious server are frequent and periodic, or because our infected device is downloading binaries from it, or because certain IRC-like or decoded orders are arriving and departing from it.
d) C&C-FileDownload – [9] This label denotes the downloading of a file to our infected device. The majority of time, this is combined with a target port or IP that is known to be a C&C server. By looking for connections with response bytes beyond 3 or 5 KB, this is found.
e) C&C – Torii – This label defines the connections as belonging to a Torii botnet. Though this botnet family is less widespread than Mirai, the categorization decision was nonetheless made using the same criteria.
f) DDoS – [10] This label indicates that the infected device is conducting a Distributed Denial of Service attack. Due to the quantity of flows aimed at the same IP address, these traffic flows are identified as being a part of a DDoS attack.
g) PartOfAHorizontalPortScan – [11] According to this label, the connections are utilized to conduct a horizontal port scan in order to acquire data for subsequent attacks. A pattern was used, in which connections used the same port, sent roughly the same

amount of data, and had a variety of IP addresses as their destinations to assign these labels.

Table 1. Label Distribution in accordance to their flows

Label	Flows
PartOfAHorizontalPortScan	93191
DDoS	14395
Benign	12849
C&C	6725
Attack	3814
C&C – Torii	16
C&C - FileDownload	12
FileDownload	2

3.2 Data Preprocessing

In the dataset considered we have mainly focused on malwares that are widespread. From the IoT-23 dataset we have used 5 captures namely Capture 3, Capture 20, Capture 34, Capture 42 and Capture 44. Each of these have both benign and attack data. The malwares in these captures are Mirai, Muhstik, Trojan and Torii. After combining all the captured data, we have pre-processed the data. Firstly, we changed the categorical data to variables 0 and 1 using pd.get_dummies and replaced '-' with 0's to have standardized data.

The data shown in the table signifies that there are no null values in our dataset. Columns such as 'ts', 'uid','id.orig_h', 'id.orig_p', 'id.resp_h', 'id.resp_p', 'service', 'local_orig', 'local_resp', 'history' that are redundant are removed. We have labels to signify the various types of attacks and benign data as shown in table 1. There are 7 attacks - Part of a horizontal scan, DDoS, C&C, Attack, C&C - Torii, C&C - FileDownload, File Download. Figure 3 is a heat map of all the features that are used in the dataset. It is used to study how each feature contributes to the overall analysis of the dataset on the various ML algorithms. After combining the data into the csv file, we built the Machine Learning models to obtain the required results.

Data splitting is an essential part of data pre-processing because it ensures the dataset is accurate when used in the creation of machine learning models. It also prevents the model from overfitting i.e., any model cannot directly predict the accuracy without being trained on a certain data. When the model is trained on some percent of the dataset, and then used for testing, the chances of obtaining a better accuracy are higher. Hence, for the purpose of this study, the dataset was split in the ratio of 4: 1 (training: testing) (Fig. 4).

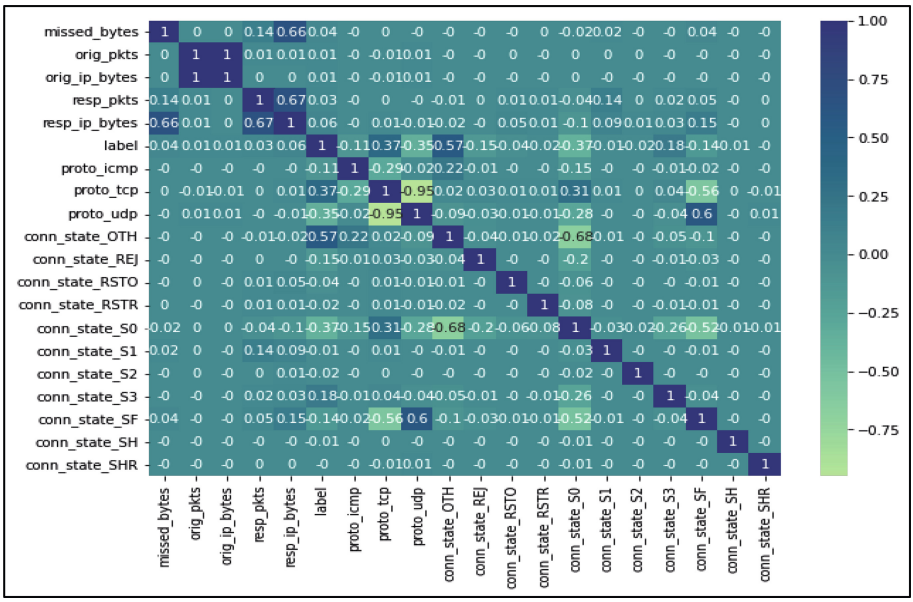

Fig. 3. Correlation between the features used in the dataset

4 Classifiers/Machine Learning Algorithms Used in the Proposed Work

Using machine learning approaches, classification is a technique that forecasts the class membership of a set of data attributes. The following machine learning algorithms were proposed to help identify and mitigate the malwares in the IoT devices.

4.1 Decision Tree

Decision Tree Algorithms is a popular and simple supervised machine learning technique. Each parent node in the decision tree must have at least one child node. Decision tree approaches can be used to tackle problems involving classification (classification trees) and regression (regression trees). The algorithm starts at the root node and attempts to predict the class of a given dataset. On the basis of the comparison, it follows the branch and jumps to the following node. It compares the value of the root with the record attribute. Up until the leaf node, comparisons are made with the sub nodes.

4.2 Naive Bayes

The Naive Bayes algorithm is a supervised learning method that uses high-dimensional training data to solve classification problems. It is a probabilistic classifier, that is, it makes predictions based on the likelihood that an object will appear. The Naive Bayes classifier operates according to the Bayes theorem's definition of conditional probability. Based on past knowledge of circumstances that might be connected to the event, the Bayes theorem determines the conditional probability of the occurrence of an event.

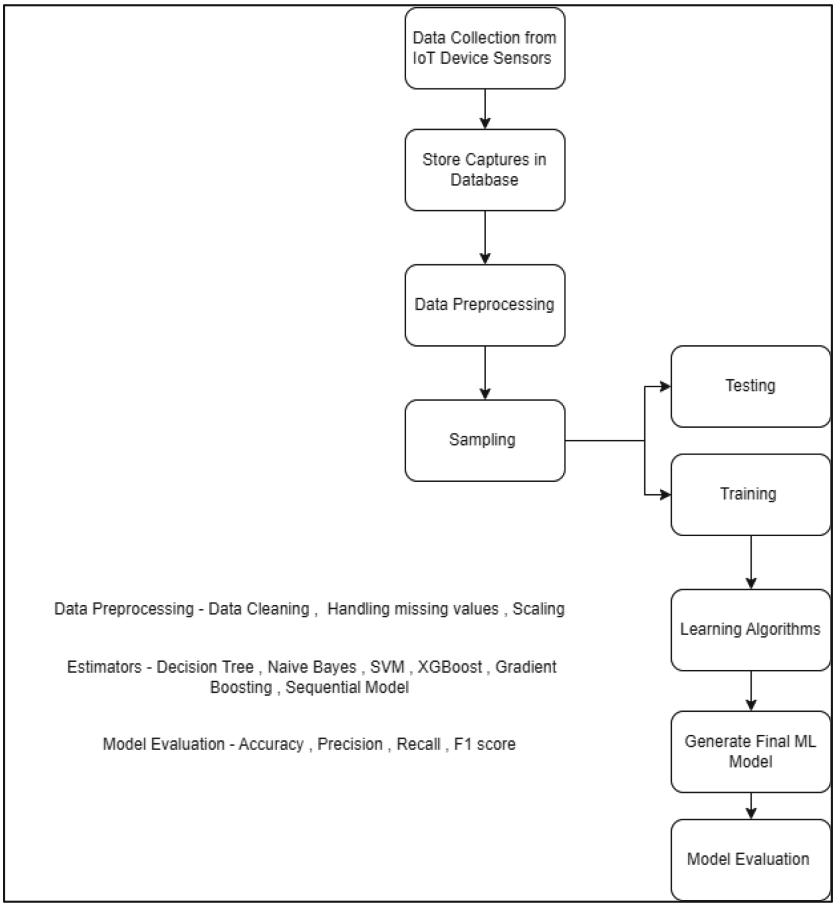

Fig. 4. Structural outline of the methodology

4.3 Support Vector Machine

This statistical learning concept-based supervised machine learning method, known as SVM, has an optimal separating hyperplane and may be used for applications such as regression and classification. Support vectors are the vectors (training data) that define the hyperplane. The SVM algorithm aims to construct the best line or decision boundary (referred to as a hyperplane) that can divide n-dimensional space into classes in order to facilitate future division. It maps data to a high-dimensional feature space in order to categorize data points even when they are not otherwise linearly separable. The data are changed to allow for the hyperplane representation of the separator once a dividing line between the categories has been found.

4.4 Convolutional Neural Network (CNN)

This particular kind of artificial neural network called a convolutional neural network (CNN) analyzes data using perceptrons, a technique for supervised learning. CNNs can be used for processing images, natural language, and other cognitive tasks. The input layer, output layer, and many hidden layers of a convolutional neural network are similar to those of other artificial neural network types. Convolutional layers use a mathematical model to transmit results to next layers. This mimics some of the activities of the visual cortex in humans. A CNN can be instantiated as a Sequential model because each layer has exactly one input and output and is stacked together to form the entire network. As we sequentially add layers to the model it is named as sequential model.

4.5 Gradient Boosting Regression

In machine learning issues including classification and regression, gradient boosting is used. Gradient boosting is the technique of "boosting" or strengthening a single weak model by combining it with a number of additional weak models to produce a more accurate indicator all together. Gradient boosting enforces the process of additively building weak models as a gradient descent approach over an objective function. Gradient boosting is an extension of boosting. Gradient boosting establishes the intended outcomes for the next model in order to reduce mistakes. It is deployed to adjust the continuous value forecasting model. It is very easy to use, can handle missing values, outliers, and features with high cardinality categorical values, and is powerful enough to identify any nonlinear relationships between your model target and features.

4.6 XGB Classifier

Extreme Gradient Boosting (XGBoost) is a distributed, gradient-boosted decision tree (GBDT) machine learning framework. This is the top machine learning library for regression, classification, and ranking problems, it offers parallel tree boosting. It was created primarily to enhance the performance and computational speed of machine learning models. Trees are constructed using XGBoost in parallel as opposed to GBDT's sequential method. It employs a level-wise approach, scanning over gradient values and assessing the quality of splits at each potential split in the training set using these partial sums [20]. In supervised machine learning, a model is trained using algorithms to discover patterns in a dataset of features and labels, and the model is then used to predict the labels on the features of a new dataset. In contrast to many other algorithms, XGBoost is an ensemble learning algorithm, it integrates the outcomes of numerous models, known as base learners, to make a prediction. Decision Trees are used by XGBoost as the base learners, just like in Random Forests [21].

5 Malwares

Malicious Software has existed for almost as long as conventional software. It was primarily restricted to PCs before the IoT. The computer security community has created strategies and equipment to combat harmful malware. IoT gadgets do not possess the complete computing capabilities on which anti-malware techniques and solutions rely. They are inexpensive, single-purpose devices with network communication capabilities. This power creates a path for attacks. IoT devices frequently utilize default usernames and passwords because there isn't a simple user interface for them. Additionally, it is not intended for them to get upgrades to software, including security updates. The malwares mentioned below successfully exploited the IoT device properties.

5.1 Mirai

The software known as Mirai, attacks consumer electronics like smart cameras and home routers and transforms them into zombie networks of remote-controlled bots. It transforms compromised equipment into a bot for DDoS attacks. It spreads to IoT devices that have default username and password settings and enabled remote access through telnet. Cybercriminals employ Mirai botnets to launch widespread distributed denial of service (DDoS) assaults against computer systems.

In general, email phishing is a clearly efficient method of computer infection. The victim is duped into either downloading an infected attachment or clicking a link that leads to a malicious website. Many times, dangerous code is constructed in such a way that it escapes detection by basic antivirus programs. In the instance of Mirai, the user only needs to leave the default username and password on a device that has just been installed unchanged, and it is due to this reason that this malware is most commonly used to attack on IOT devices.

Mirai is split into three halves. The CNC server gives botnet members a virtual terminal, maintains a record of the registered bots, and transmits attack commands to the bots. On reported susceptible systems, the loader transfers and runs malware. A bot looks for weak targets and launches a DoS attack whenever necessary [2].

5.2 Torii

Torii is different from Mirai and other known botnets, especially in the sophisticated methods it employs. As opposed to other IoT botnets, this one seeks to remain hidden and persistent after a device has been infected. It also avoids typical botnet activities like DDOS attacks against other connected devices and cryptocurrency mining.

It has a wide range of functions for the exfiltration of private data and a retrieval-capable modular design. It is capable of carrying out orders and executables via several encrypting communication levels. Additionally, it can spread to a wide range of hardware, including x86, x64, MIPS, PowerPC, ARM, and many more [17].

5.3 Trojan

A Trojan, sometimes known as a Trojan horse, is a form of malware that hides its true purpose in order to trick a user into believing it to be a harmless programme. Trojans are used as a delivery system for a number of different types of malwares. It is intended to hurt, disrupt, steal, or harm your data or network. It accomplishes this by reading passwords, capturing keyboard strokes, or opening the way for more spyware that can take over the entire machine. These actions can include: Deleting data, blocking data, modifying data, copying data, interrupting the functioning of computers or computer networks.

It attempts to trick the user into installing and running malware on their device. Once implanted, a Trojan can carry out the intended function. Unlike computer viruses, these cannot replicate themselves.

Generally, an IoT device is a combination of several parts. Memory, firmware, the physical interface, the web interface, and network services are examples of components where vulnerabilities may exist in a device. Attacks may come through the channels that link different IoT components together. As a result, trojan malware is frequently used as software or hardware components on IoT devices for assaults.

5.4 Mushtik

Mushtik botnet infiltrates IoT devices, such as routers, using well-known web application flaws to mine cryptocurrency using open-source tools like XMRig and cgminer. For its command-and-control (C2) operations, it uses IRC servers.

A Mushtik attack is carried out in various phases. Firstly, from the attacker's server a payload file with the name "pty" and a number is downloaded. Mushtik will connect to the IRC channel after a successful installation in order to accept orders. The C2 infrastructure that powers the Mushtik botnet is provided by IRC servers. Mushtik will be prompted to download both a scanning module and an XMRrig miner. By focusing on other Linux servers and home routers, the scanning module is utilized to expand the botnet. The settings of Mushtik's payload and scanning module are encrypted using single-byte XOR using the Mirai source code [18]. The Mushtik botnet, which can spread itself like a worm and attack Linux servers and IoT devices [19].

6 Result Discussion

The proposed algorithms were compared with the existing Machine learning models and hence demonstrated in Table 2. Using Decision Tree, Support Vector machine and Naive bayes, Gotsev et al. [22] obtained an accuracy of 1.00, 0.74 and 0.58 respectively. These algorithms were deployed in 2021. Nicolas Stoian [7] implemented the Naive bayes classifier in the year 2020 however it did poorly because it only managed to attain an accuracy of 0.23. Chunduri et al.'s [24] usage of the SVM classifier and the gradient boosting algorithms resulted in accuracy rates of 0.9472 and 0.9936, respectively, in the year 2021. Accuracy rates of Support Vector Machine classifier was reported to be 0.8352, according to the study by Strecker et al. [25] in the year 2021.

Table 2. Comparison with previous studies for evaluating ML model

Model	Study	Accuracy
Naïve Bayes	[22]	0.58
	[7]	0.23
	This Study	0.934
Support Vector Machine	[22]	0.74
	[24]	0.9472
	[25]	0.8352
	This Study	0.934
Decision Tree	[22]	1.00
	This study	0.97
Gradient Boosting Machine	[24]	0.9936
	This study	0.777
XGBoost	This study	0.974
Sequential Model	This study	0.943

The proposed Machine Learning algorithms were trained and tested with the chosen dataset, IoT - 23. The dataset was tested on six different algorithms and they were evaluated on the metric of their accuracy score. Naïve Bayes classifier is a probabilistic algorithm that makes use of Bayes' theorem with robust independent assumptions. With the chosen dataset, this algorithm shows an accuracy of 93.4%. Decision Tree classifier is a supervised learning technique that makes a decision based on a certain set of rules. This algorithm with the chosen dataset, showed an accuracy score of 97.04%. Gradient Boosting algorithm processes data by merging together many weak models to create a strong robust model. This algorithm guarantees an accuracy of 77.75%. XGBoost is a scalable and extremely accurate gradient boosting solution that pushes the limits of computing power for boosted tree algorithms. With the dataset chosen, this algorithm showed an accuracy of 97.4%. Support Vector Machine classifier (SVM) aims to construct the best line or decision boundary (referred to as a hyperplane) that can divide n-dimensional space into classes in order to facilitate future division. On being implemented with the chosen dataset, this algorithm shows an accuracy of 93.41%. CNN classifier is a type of neural network classifier that is used to identify a specific pattern in a dataset, which can then be used to identify a malware in the system. This classifier with the chosen dataset, gives an accuracy of 94.4%. The XGBoost algorithm hence shows the highest most accurate result with the chosen dataset, and will therefore most accurately identify the malwares in the dataset chosen (Fig. 5).

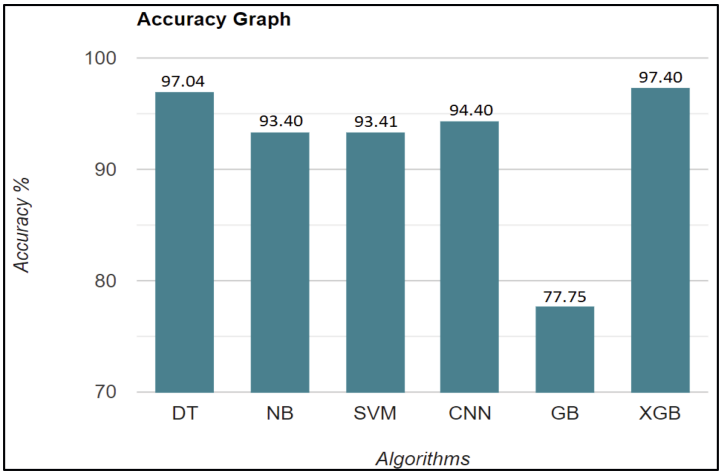

Fig. 5. Accuracy Graph for the Algorithms Implemented

7 Conclusion and Future Work

There has been an upward surge in the prevalence of IoT devices in various sectors. The increasing dependency on these devices is what makes them more vulnerable to assault launched by cyber criminals. It hence becomes important to devise ways to mitigate these security risks and prevent cyber-attacks. The current technology being used to identify and deviate attacks are inefficient to find the advanced malwares present on these IoT devices. This work aims at working on malware evaluation i.e., an approach or study that identifies the supply, functioning, and capacity outcomes of a malware specimen. We have also proposed 6 different algorithms, namely: Decision Tree Classifier, Gaussian Naive Bayes, Gradient boosting algorithm, XGBoost, SVM Classifier and CNN classifier. These algorithms were implemented on a chosen dataset, and have demonstrated better results for the newer IoT datasets. XGBoost proved to be the most accurate among all the algorithms, with an accuracy of 97.4%.

The researchers can extensively perceive the behavior of varied varieties of malware attacks in IoT in the future. A Wider more varied dataset can be used to analyze the presence of malwares in IoT devices. Algorithms like Adboost can furthermore be implemented and be made to give a more accurate result with the different IoT datasets available. Newer forms of Machine Learning techniques involving neural networks and ensemble methodologies can be implemented to prevent IoT devices from being vulnerable to malwares and other forms of security risks.

References

1. Servida, F., Casey, E.: IoT forensic challenges and opportunities for digital traces. Digit. Investig. **28**(Supplement), S22–S29 (2019). https://doi.org/10.1016/j.diin.2019.01.012. ISSN 1742-2876

2. Sinanović, H., Mrdovic, S.: Analysis of Mirai malicious software. In: Proceedings of the 2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), pp. 1–5 (2017). https://doi.org/10.23919/SOFTCOM.2017.8115504
3. Chen, S., Xu, H., Liu, D., Hu, B., Wang, H.: A vision of IoT: applications, challenges, and opportunities with China perspective. IEEE Internet Things J. **1**(4), 349–359 (2014). https://doi.org/10.1109/JIOT.2014.2337336
4. Xiao, L., Wan, X., Lu, X., Zhang, Y., Wu, D.: IoT security techniques based on machine learning: how do IoT devices use AI to enhance security? IEEE Signal Process. Mag. **35**(5), 41–49 (2018). https://doi.org/10.1109/MSP.2018.2825478
5. Lee, Y.-T., et al.: Cross platform IoT-malware family classification based on printable strings. In: Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 775–784 (2020). https://doi.org/10.1109/TrustCom50675.2020.00106
6. Sahu, A.K., Sharma, S., Tanveer, M., Raja, R.: Internet of Things attack detection using hybrid deep learning model. Comput. Commun. **176**, 146–154 (2021). https://doi.org/10.1016/j.comcom.2021.05.024.ISSN0140-3664
7. Stoian, N.-A.: Machine learning for anomaly detection in IoT networks: malware analysis on the IoT-23 data set (2020)
8. Dutta, V., Choraś, M., Pawlicki, M., Kozik, R.: Detection of cyberattacks traces in IoT data. J. Univ. Comput. Sci. **26**, 1422 (2020). https://doi.org/10.3897/jucs.2020.075
9. Bederna, Z., Szadeczky, T.: Cyber espionage through Botnets. Secur. J. **33**, 43–62 (2020). https://doi.org/10.1057/s41284-019-00194-6
10. Salim, M.M., Rathore, S., Park, J.H.: Distributed denial of service attacks and its defenses in IoT: a survey. J. Supercomput. **76**, 5320–5363 (2020). https://doi.org/10.1007/s11227-019-02945-z
11. Bhuyan, M., Bhattacharyya, D.K., Kalita, J.K.: Surveying port scans and their detection methodologies. Comput. J. **54**(10), 1565–1581 (2011). https://doi.org/10.1093/comjnl/bxr035
12. Ullah, I., Mahmoud, Q.H.: Design and development of a deep learning-based model for anomaly detection in IoT networks. IEEE Access **9**, 103906–103926 (2021). https://doi.org/10.1109/ACCESS.2021.3094024
13. Ullah, I., Ullah, A., Sajjad, M.: Towards a hybrid deep learning model for anomalous activities detection in Internet of Things networks. IoT **2**(3), 428–448 (2021). https://doi.org/10.3390/iot2030022
14. Booij, T.M., Chiscop, I., Meeuwissen, E., Moustafa, N., den Hartog, F.T.H.: ToN_IoT: the role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. IEEE Internet Things J. **9**(1), 485–496 (2022). https://doi.org/10.1109/JIOT.2021.3085194
15. Nascita, A., Cerasuolo, F., Di Monda, D., Garcia, J., Montieri, A., Pescapè, A.: machine and deep learning approaches for IoT attack classification (2022). https://doi.org/10.1109/INFOCOMWKSHPS54753.2022.9797971
16. Dutta, V., Choraś, M., Pawlicki, M., Kozik, R.: A deep learning ensemble for network anomaly and cyber-attack detection. Sensors **20**(16), 4583 (2020). https://doi.org/10.3390/s20164583
17. Kroustek, J., Iliushin, V., Shirokova, A., Neduchal, J., Hron, M.: Torii botnet-not another mirai variant (2020). https://blog.avast.com/new-torii-botnet-threat-research
18. Chinese-linked Mushtikbotnet targets Oracle WebLogic, Drupal. BleepingComputer. https://www.bleepingcomputer.com/news/security/chinese-linked-muhstik-botnet-targets-oracle-weblogic-drupal/. Accessed 28 Oct 2022
19. MushtikBotnet attacks tomato routers to harvest new IoT devices. Unit42, 21 January 2020. https://unit42.paloaltonetworks.com/muhstik-botnet-attacks-tomato-routers-to-harvest-new-iot-devices/

20. Nvidia, "What is XGBoost?" NVIDIA data science glossary. https://www.nvidia.com/en-us/glossary/data-science/xgboost/
21. B. T: Beginner's Guide to XGBoost for Classification Problems. Medium, 12 October 2021. https://towardsdatascience.com/beginners-guide-to-xgboost-for-classification-problems-50f75aac5390
22. Gotsev, L., Dimitrova, M., Jekov, B., Kovatcheva, E., Shoikova, E.: A cybersecurity data science demonstrator: machine learning in IoT network security, p. 6 (2021)
23. Vitorino, J., Andrade, R., Praça, I., Sousa, O., Maia, E.: A comparative analysis of machine learning techniques for IoT intrusion detection. In: Aïmeur, E., Laurent, M., Yaich, R., Dupont, B., Garcia-Alfaro, J. (eds.) Foundations and Practice of Security. FPS 2021. Lecture Notes in Computer Science, vol. 13291. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08147-7_13
24. Chunduri, H., Gireesh, T., Charan, P.V.S.: A multi class classification for detection of IoT Botnet malware. In: Chaubey, N., Parikh, S., Amin, K. (eds.) COMS2 2021. CCIS, vol. 1416, pp. 17–29. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-76776-1_2
25. Strecker, S., Dave, R., Siddiqui, N., Seliya, N.: A modern analysis of aging machine learning based IoT cybersecurity methods. J. Comput. Sci. Appl. **9**(1), 16–22 (2021). https://doi.org/10.12691/jcsa-9-1-2
26. Zolanvari, M., Teixeira, M.A., Gupta, L., Khan, K.M., Jain, R.: Machine learning-based network vulnerability analysis of industrial Internet of Things. IEEE Internet Things J. **6**(4), 6822–6834 (2019). https://doi.org/10.1109/JIOT.2019.2912022
27. Jan, S.U., Ahmed, S., Shakhov, V., Koo, I.: Toward a lightweight intrusion detection system for the Internet of Things. IEEE Access **7**, 42450–42471 (2019). https://doi.org/10.1109/ACCESS.2019.2907965