Intrusion Detection System using Machine Learning Models

Aditya Kumar
IIITM Gwalior
Gwalior, India
mtics_202303@iiitm.ac.in

Ilapavuluri Raviteja Srikouthsav
IIITM Gwalior
Gwalior, India
mtics_202309@iiitm.ac.in

Kanishka Singh
IIITM Gwalior
Gwalior, India
mtics_202310@iiitm.ac.in

***Abstract*— The Internet of Things (IoT) has emerged as a rapidly expanding technology that facilitates the connectivity of various devices and systems, allowing them to exchange data and interact seamlessly. However, alongside its benefits, IoT presents significant security challenges, particularly concerning the protection of sensitive data and the prevention of unauthorized access or attacks. The widespread adoption of IoT technology has led to an increase in connected devices and systems, expanding the potential attack surface and making it easier for attackers to exploit vulnerabilities. Additionally, the data generated by IoT devices often contains sensitive information, necessitating robust security measures to safeguard against unauthorized access or misuse. Intrusion detection systems play a crucial role in mitigating potential threats in real-time, but further research is needed to develop effective security measures tailored to the unique challenges posed by IoT networks. A proposed framework offers a promising approach to addressing these security challenges, particularly in enhancing intrusion detection systems for IoT devices, but further exploration is required to evaluate its generalizability and performance in real-world scenarios.**

Keywords— IoT security, malware attack, exploit vulnerabilities, ML framework, IoT devices.

I. INTRODUCTION

Internet of Things (IoT) is a term used to describe a network of intelligent gadgets that can connect and share data with other systems and devices over the internet. These can be anything from sophisticated automation systems for the home to incredibly expensive and delicate machinery utilized in industries. Around 10 billion IoT devices are expected to be in use right now. All of these gadgets together create a sizable attack surface that could be used for malicious attacks. [13]
Some of the more frightening vulnerabilities found on IoT devices have brought IoT security further up the stack of issues that need to be addressed quickly. A number of botnets, such as the infamous Mirai botnet and Bashlite botnet, pose a significant threat to users as they can leverage a vast number of such devices to coordinate attacks. We aim to develop a solution that can be deployed onto existing IoT networks as botnets have been used to launch Distributed Denial-of-Service (DDoS) attacks affecting critical Internet infrastructure.
The goal of sharing the data and connecting to other networks has been effectively achieved since IoT technology uses a wide variety of devices, including sensors, processors, and many other technologies. The shared data may not be secure because of the numerous related vices, which poses security concerns. IoT security refers to the safeguarding of data exchanged between various networks by means of IoT devices and IoT technologies.

II. LITERATURE SURVEY

We went through other works in the same field to understand which approaches had already been taken by people and how their approaches fared. For generating the network dataset, a honeypot network has been built to simulate an IOT device [1,2]. This approach was preferred since it provided a chance for new malware to show up in the log files that may not be present in previous literature [1,2]. The features were extracted from the log files generated and were used to run classifiers using a feature set similar to something one would get if they run Zeek (formerly bro) on the captured traffic. We referred to surveys that covered the different approaches towards IDS that were undertaken by people [2]. It was a comprehensive review of various strategies used in intrusion detection, and their deployment methods, including information on the datasets available.

It highlighted the need to have a robust mechanism to classify intrusion in the IoT network since not all attacks can be recognized by the signature matching mechanism of traditional IDS systems. It showed key differences between the various approaches along with respective advantages and disadvantages.

There are a lot of traditional defense mechanisms employed to mitigate attacks against IoT devices such as filtering packets through the use of Firewall and proxies, adopting encryption standards, use of robust authentication systems and log and auditing of the associated network activity. Since IoT devices are generally lightweight devices, a lot of them cannot have such mechanisms.[16]

It has been noted that traditional IDS based on signature matching is highly effective but lacks the required tangibility to deal with changing threats. Thus, a technique that adapts itself to the changes in attacks (learns through experience).

Shafiq, Tian, Sun, et al., 2020 [5] used features in the BOT-IOT dataset [17] to find a framework for the detection of malicious attacks. Their work emphasized that working with a large number of features in a dataset adds extra overhead to the classification process and ends up being a barrier when efficient classification frameworks are required. They argued that correct and nonredundant feature selection was an imperative part of modeling the dataset. They have proposed

a framework for robust feature selection which works in 7 phases. In the end, they merged the optimized features they got through various phases into a large feature set. Four different classifiers were run on this feature set and the results were seen as promising.

Operation Codes (Opcodes) were also used as possible features for malware detection.[14] An Opcode is a piece of instruction that is executed by the CPU. It basically describes the behavior of an executable file. In assembly language, an opcode is a command such as CALL, ADD, or MOV. Based on this feature, they proposed a method using Recurrent Neural Network (RNN) deep learning to detect IoT malware using Opcode sequences. They used three LSTM models with different configurations.

Visual approaches to classify an incoming network have also been studied which involves turning pcap files into an image. This type of malware analysis falls under the category of static analysis.

Binvis is a very popular tool used by security researchers to get a visual representation of binary-type files [11]. Binvis works by comparing the byte values in the pcap file to their respective ASCII values and colors them according to a predefined color scheme (printable ASCII were coded as blue, control bytes as green, extended bytes as red, and white were coded to null and non-breaking spaces). The image is mapped using space-filling curves to ensure that data points which are originally close remain close in the output image.

The images were then fed to the ResNet50 neural network which is used for Image Classification. The network pcaps were obtained from Netresec and the IoT 23 dataset.

Garcia and Muga[4] used the Malimg malware dataset[10].The dataset was generated by converting the malware binaries into groups of 8-bit vectors. These were then used to produce grayscale images and the images were then used for training models.

Usually, IoT devices communicate via machine-to-machine protocols such as Message Queuing Telemetry Transport (MQTT). Due to the heterogeneous structure of IoT and the absence of security by design methodologies, security mechanisms in environments with MQTT traffic are needed, and they can be deployed as Intrusion Detection Systems (IDS).

There are also Deep Learning (DL) based Network IDS models trained using a public dataset containing MQTT attacks [12]. To guarantee the reproducibility of their proposal, they trained the DL model, embedded in the IDS, using the public dataset MQTT-IoT-IDS2020, where IoT devices use the MQTT protocol.

In the thesis by Michael Austin [6] IOT 23 dataset [7] was used along with different classifiers such as Naive Bayes, SVM with a linear kernel and other models.

After browsing the previous works, we have come to the conclusion that although signature matching for malware is one of the most common techniques for detecting malware, it cannot stop all attacks due to the dynamic nature of threats. To tackle this scenario where the threats are ever-changing, IDS systems built using learning techniques are a better match for it.

We also saw entropy-based measures for the detection of anomalies in network traffic. These overcome the limitations of traditional IDS systems as they provide more flexibility when dealing with new classes of malware. These generally work by defining a baseline activity for the network and labeling network flows that are too different from this baseline as anomalous. The paper measured the deviation between the profiles of normal traffic and incoming flow records using their own entropy function which has been described in the paper [15].

III. METHODOLOGY

Several machine learning algorithms have been employed on the IoT-23 dataset to demonstrate the effectiveness of the approach.

- *Machine Learning Algorithms*

The following algorithms have been used as a classifier model for intrusion detection systems: Naïve Bayes, Decision Tree, Random Forest, Stochastic Gradient Descent, Linear Support Vector Machine.

Naive Bayes is a type of probabilistic algorithm commonly used for classification. It is based on Bayes' theorem, which is a mathematical formula for calculating conditional probabilities. In the context of classification, the Naive Bayes algorithm uses the probabilities of each attribute belonging to each class to make a prediction. Naive Bayes is a simple but powerful algorithm for predictive modeling. It is based on the idea that the value of a particular feature is independent of the value of any other feature, given the class variable. This assumption is called the "naive" assumption because it is often unrealistic in real-world scenarios. However, despite this assumption, the algorithm often performs very well in practice. The formula for the Naive Bayes algorithm is as follows:

$$P(A|B) = P(B|A) * P(A) / P(B).$$

Here, P(A|B) is the probability of A given B, P(B|A) is the probability of B given A, P(A) is the probability of A, and P(B) is the probability of B.

Stochastic Gradient Descent (SGD) is a popular optimization algorithm often used in machine learning to train models on large datasets. It uses randomness to make predictions and iteratively improves the model by moving along with the negative gradient of the loss function. This means that at each step, the algorithm calculates the gradient of the loss function concerning the model's parameters and updates the parameters in a way that reduces the loss. By doing this repeatedly, the algorithm can find the parameters that minimize the loss and produce the most accurate predictions. The formula for the SGD algorithm is as follows:

$$w(t+1) = w(t) – learning\_rate * gradient(w(t))$$

Here, w(t) is the parameter vector at time step t, learning_rate is a hyperparameter that determines the step size at each iteration, and gradient(w(t)) is the gradient of the objective function with respect to the parameter vector w at time step t.

Random forests are a good choice for classification tasks because they can handle high-dimensional and complex data well. This is because the individual decision trees in a random forest are trained using different subsets of the features, and

the final prediction is made by considering the collective predictions of all the trees. This allows the algorithm to find complex patterns in the data that may be difficult to identify using a single decision tree. An advantage of random forests is that they can help to reduce overfitting, which is a common problem in decision tree models. This makes it a good choice for the IoT 23 data since it is a large dataset with high chances of overfitting.

A random forest is an ensemble learning method that consists of a collection of decision trees. The idea behind a random forest is to train many decision trees on random subsets of the data, and then average their predictions to make a final prediction. This process of training multiple models and combining their predictions is called "ensemble learning".

Linear SVMs are an excellent choice for classification because they can perform well even with high-dimensional or complex data. This is because the SVM algorithm only considers the points closest to the decision boundary (known as support vectors) when making predictions, so it is not affected by the majority of the data points that are further away from the boundary. This allows it to focus on the most critical information in the data and make accurate predictions. Another advantage of linear SVMs is that they are relatively easy to implement and computationally efficient, so they can be used to train models on large datasets. Overall, linear SVMs are versatile and practical tools for classification tasks. The formula for the linear SVM is as follows:

$f(x) = w * x + b$

Here, f(x) is the predicted class for a given input x, w is the weight vector, and b is the bias. The weight vector and bias term are learned from the training data using an optimization algorithm such as gradient descent.

- *Performance Metrics*

The below metrics were used to measure the performance of the classifiers on the dataset.

• Accuracy = (True Positive + True Negative) / (True Positive + False Negative + False Positive + True Negative)

• Probability of Detection = True Positive / (True Positive + False Negative)

• Probability of False Alarm = False Positive / (False Positive + True Negative)

Our workflow consisted of the network traffic which has been captured as a pcap file being routed through Zeek. Zeek generates a log file which is then preprocessed and sent to our Machine Learning model for classification. The model was then used to classify the incoming network traffic as malicious or benign.

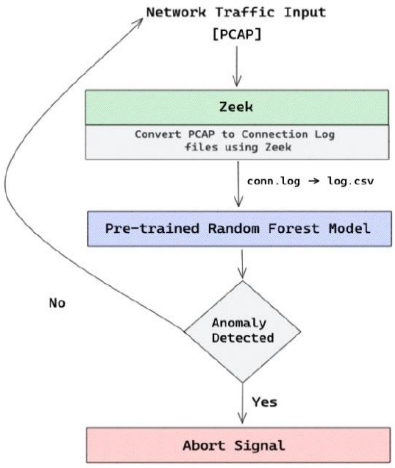

Figure 1: IDS Process

IV. DATASET

The training of our ML model was done on three publicly available datasets i.e. IOT 23 and UNSW NB15 dataset.

A. IoT 23 Dataset

It is by far the most comprehensive and up-to-date dataset available. It was prepared by simulating attacks using various malware/botnets on real-world devices: a Philips HUE smart LED lamp, an Amazon Echo home intelligent personal assistant, and a Somfy smart door lock. This will be the primary dataset for our analysis/training purposes. In 2020, Parmisano, Garcia, and Erquiaga published the IoT-23 dataset, which includes packet captures and Zeek logs from twenty malware and three benign traffic samples.
A Raspberry Pi was used to simulate an IoT device, and different malware was run on it. Both malicious and benign scenarios were run in a controlled network environment with unrestricted internet connection like any other IoT device. An Amazon Echo, a Philips HUE retail LED light bulb, and a Somfy smart lock were among the IoT devices used. The dataset consists of 23 scenarios of different IoT traffic. Each scenario corresponds to a different network capture from an infected IoT device running a specific malware. There are three scenarios of regular IoT traffic, with 20 scenarios being infected. The logs generated by Zeek were obtained by running the Zeek network analyzer on the PCAP files captured from the infected IoT devices. The traffic was captured over 24 hours, with one capture running for 112 hrs. The dataset provides both network PCAPs from the infected

B. UNSW NB15 dataset

The UNSW 15 IoT dataset is a collection of network traffic data from Internet of Things (IoT) devices. It was created by researchers at the University of New South Wales as a benchmark dataset for evaluating intrusion detection systems for IoT devices. The dataset contains a mix of normal and attack behavior, and it has been widely used in research studies related to IoT security. The dataset contains a total of 9,000,000 records, with each record representing a network flow and its corresponding labels indicating normal or attack

behavior. The records are divided into two sets: the training set, which contains 4,900,000 records, and the test set, which contains 4,100,000 records. The training set is used to train machine learning models, while the test set is used to evaluate the performance of the trained models. The UNSW 15 IoT dataset has been widely used in research studies on intrusion detection for IoT devices, and it has been shown to provide a challenging testbed for evaluating the performance of different machine learning algorithms. It is a valuable resource for researchers working in the field of IoT security, and it has contributed to the advancement of machine learning-based intrusion detection systems for IoT devices. We have used this dataset for checking the validity of our trained ML models (trained using the IoT 23 dataset).

V. EXPERIMENT AND RESULTS

In this section, we propose a ML model for intrusion detection. We had a total of 23 scenarios, with 20 scenarios containing malicious activity and three being completely benign. The benign scenarios are Honeypot captures of devices such as Somfy Door Lock, Philips HUE, and Amazon Echo.

A. Preprocessing of Data set

We used all 23 scenarios and concatenated them in a single data frame. We only took 100,000 records from each scenario, which combined a total of 1,646,508 records. The number of features in our combined dataset numbered 23 are shown in the figure 2.

We dropped features such as uid, id.orig_h, id.resp_h, and detailed-label. uid is a unique identifier for the network flow . The features id.orig_h, id.resp_h are addresses of originating and responding endpoints therefore, they are dropped because always identifying particular addresses as malicious will end up being a simple blacklist. Finally, the detailed-label was removed because it further specified the type of malicious activity, which was not needed by us.

```
1   ts
2   uid
3   id.orig_h
4   id.orig_p
5   id.resp_h
6   id.resp_p
7   proto
8   service
9   duration
10  orig_bytes
11  resp_bytes
12  conn_state
13  local_orig
14  local_resp
15  missed_bytes
16  history
17  orig_pkts
18  orig_ip_bytes
19  resp_pkts
20  resp_ip_bytes
21  tunnel_parents
22  label
23  detailed-label
```

Figure 2: Features in Dataset

B. Performance evaluation of machine learning algorithms

Four supervised ML algorithms, namely, Naive Bayes, Stochastic Gradient Descent, Random Forest, Linear SVM have been implemented and evaluated for their performance to detect intrusion

The performance of these algorithms is given in the form of metrics, i.e., accuracy, precision, recall, and F-score in Figure 3.

We deduce that the Random Forest method performed better for intrusion detection.

C. Detection process

The trained model would be deployed in the IoT network of interest. Copy of network traffic of the nodes would be sent to the server (model) in frequent intervals. In case an intrusion is detected, an abort signal would be sent to turn off the attacked node to prevent further spreading of the attack.

Model	Time to train(s)	Accuracy (%)	Precision (%)	Recall (%)	F1-Score
GNB	0.731	87.837	72.929	50.026	59.344
SGD	2.748	87.835	43.917	50	46.761
Linear SVM	265.230	88.835	44.328	51.2	47.516
Random Forest	249.895	98.629	99.273	98.766	99.018

Figure 3: Performance metrics

D. Testing on UNSW NB15 dataset

After our model underwent rigorous training on the IoT 23 dataset, a comprehensive examination was conducted to ensure its efficacy. This involved subjecting the model to various publicly accessible datasets to validate its performance. Among these datasets, the UNSW NB15 emerged as a prominent choice due to its comprehensive collection of malicious network captures across diverse scenarios. Through meticulous testing, our findings unequivocally demonstrated the proficiency of our model in accurately classifying the UNSW NB15 dataset as malicious across all depicted scenarios.

VI. CONCLUSION AND FUTURE WORK

The project developed a highly accurate Intrusion Detection System (IDS) for IoT networks, blending machine learning and signature-based analysis. Extensive testing with diverse datasets, including IoT 23 Dataset and UNSW NB-15 Dataset, verified its efficacy in distinguishing between benign and malicious traffic. The IDS successfully detected known threats using signatures and exhibited capability in identifying unknown threats, including zero-day exploits. A proof of concept (POC) using a Raspberry Pi simulated IoT network gateway showcased the system's effectiveness in

real-world scenarios. Overall, the project represents a significant advancement in IoT security, promising substantial real-world impact and offering avenues for further enhancement.

Proposed improvements to the existing Intrusion Detection System (IDS) include:

1. Enhanced Malware/Attack Classification: Utilizing machine learning algorithms to classify malware types and attack methods based on network traffic patterns, aiding in precise identification and countermeasure selection.

2. Centralized Server for Multiple IoT Networks: Implementing a single server to manage multiple IoT networks' intrusion detection tasks, optimizing resource usage and simplifying administration, particularly beneficial for smaller IoT networks like home setups.

3. Adoption of Deep Learning Models: Incorporating deep learning models to enhance IDS capabilities by capturing intricate patterns in network data, leveraging the complexity of artificial neural networks to detect anomalies more effectively than traditional machine learning approaches.

4. Cross-Training of Machine Learning Models: Improving anomaly-based IDS by integrating data from signature-based IDS for additional training, and vice versa, facilitating a collaborative approach to enhance detection accuracy and adaptability to evolving threats.

VII. REFERENCES

[1] Ruchi Vishwakarma and Ankit Kumar Jain. "*A Honeypot with Machine Learning based Detection Framework for defending IoT based Botnet DDoS Attacks*". In: 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI). Apr. 2019, pp. 1019–1024. doi: 10.1109/ICOEI.2019.8862720.

[2] Khraisat, A., Alazab, A. A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecur 4, 18 (2021). https://doi.org/10.1186/s42400-021-00077-7

[3] G. Bendiab, S. Shiaeles, A. Alruban and N. Kolokotronis, "*IoT Malware Network Traffic Classification using Visual Representation and Deep Learning*," 2020 6th IEEE Conference on Network Softwarization (NetSoft), 2020, pp. 444-449, doi: 10.1109/NetSoft48620.2020.9165381.

[4] Felan Carlo C Garcia and Felix P Muga Ii. "*Random Forest for Malware Classification*". In: arXiv preprint arXiv:1609.07770 (), p. 4.

[5] Shafiq, M., Tian, Z., Sun, Y., Du, X., and Guizani, M. Selection of effective machine learning algorithms and Bot-IoT attacks traffic identification for internet of things in smart city. Future Generation Computer Systems 107 (jun 2020), 433–442.

[6] Austin, Michael, "*IoT Malicious Traffic Classification Using Machine Learning*" (2021). Graduate Theses, Dissertations, and Problem Reports. 8024.

[7] Sebastian Garcia, Agustin Parmisano, & Maria Jose Erquiaga. (2020). IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.4743746

[8] N. Moustafa et al., "*Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network dataset)*," in MilCIS, IEEE, 2015.

[9] Hanan Hindy, Christos Tachtatzis, Robert Atkinson, Ethan Bayne, Xavier Bellekens, June 23, 2020, "*MQTT-IoT-IDS2020: MQTT Internet of Things Intrusion Detection Dataset*", IEEE Dataport, doi: https://dx.doi.org/10.21227/bhxy-ep04.

[10] Nataraj, Lakshmanan & Karthikeyan, Shanmugavadivel & Jacob, Grégoire & Manjunath, B.. (2011). Malware Images: Visualization and Automatic Classification.10.1145/2016904.2016908.

[11] Visual analysis of binary files. Online: url https://binvis.io/

[12] F. Mosaiyebzadeh, L. G. Araujo Rodriguez, D. Macêdo Batista and R. Hirata, "*A Network Intrusion Detection System using Deep Learning against MQTT Attacks in IoT*," 2021 IEEE Latin-American Conference on Communications (LATINCOM), 2021, pp. 1-6, doi: 10.1109/LATINCOM53176.2021.9647850.

[13] Introduction to IoT by Sudeppto Mishra

[14] HaddadPajouh, H., Dehghantanha, A orcid.org/0000-0002-9294-7554, Khayami, R. et al. (1 more author) (2018) A Deep Recurrent Neural Network Based Approach for Internet of Things Malware Threat Hunting. Future Generation Computer Systems, 85. pp. 88-96. ISSN 0167-739X https://doi.org/10.1016/j.future.2018.03.007

[15] Jérôme François, Cynthia Wagner, Radu State, Thomas Engel. SAFEM: Scalable analysis of flows with entropic measures and SVM. Network Operations and Management Symposium, Apr 2012, Lahaina, United States. pp.510-513, ff10.1109/NOMS.2012.6211943ff. ffhal-00734967

[16] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac and P. Faruki, "*Network Intrusion Detection for IoT Security Based on Learning Techniques*," in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671-2701, third quarter 2019, doi: 10.1109/COMST.2019.2896380.

[17] Nour Moustafa, October 16, 2019, "*The Bot-IoT dataset*", IEEE Dataport, doi: https://dx.doi.org/10.21227/r7v2-x988

[18] Booij, Tim M., et al. "*ToN_IoT: The Role of Heterogeneity and the Need for Standardization of Features and Attack Types in IoT Network Intrusion Datasets*." IEEE Internet of Things Journal (2021).

[19] Sánchez, Pedro Miguel Sánchez, et al. "*A Survey on Device Behavior Fingerprinting: Data Sources, Techniques, Application Scenarios, and Datasets*." IEEE Communications Surveys & Tutorials (2021).

[20] Sahu, Amiya Kumar, et al. "*Internet of Things attack detection using hybrid Deep Learning Model*." Computer Communications (2021).

[21] Ahmad, Rasheed, and Izzat Alsmadi. "*Machine learning approaches to IoT security: A systematic literature review*." Internet of Things (2021): 100365.

[22] Dutta, Vibekananda, et al. "*A deep learning ensemble for network anomaly and cyber-attack detection*." Sensors 20.16 (2020): 4583.

[23] Chunduri, Hrushikesh, T. Gireesh Kumar, and PV Sai Charan. "*A Multi Class Classification for Detection of IoT Botnet Malware*." International Conference on Computing Science, Communication and Security. Springer, Cham, 2021.

[24] Dutta, Vibekananda, et al. "*Hybrid model for improving the classification effectiveness of network intrusion detection*." Conference on Complex, Intelligent, and Software Intensive Systems. Springer, Cham, 2020.

[25] Zeek: https://docs.zeek.org/en/master/

[26] Suricata: https://suricata.io/documentation/

[27] Manos Antonakakis et al. "*Understanding the Mirai Botnet*". In: 26th USENIX Security Symposium, p. 19.Konrad Rieck, Patrick Stewin, and Jean-Pierre Seifert, eds. Detection of Intrusions and Malware, and Vulnerability Assessment. Red. by David Hutchison et al. Vol. 7967. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, isbn: 978-3-642-39235-1. doi: 10 . 1007 / 978 - 3 - 642 - 39235 - 1.

[28] Ruchi Vishwakarma and Ankit Kumar Jain. "*A Honeypot with Machine Learning based Detection Framework for defending IoT based Botnet DDoS Attacks*". In: 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI). 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI). Apr. 2019, pp. 1019–1024. doi: 10.1109/ICOEI.2019.8862720.

[29] Anand Ravindra Vishwakarma. "*Network Traffic Based Botnet Detection Using Machine Learning*". In: SJSU Master's Projects 917 (), p. 67.

[30] Nicolas-Alin Stoian. "*Machine Learning for Anomaly Detection in IoT networks: Malware analysis on the IoT-23 Data set*". In: Bachelor's thesis, University of Twente (), p. 10.

[31] sklearn. 1.5. Stochastic Gradient Descent — scikit-learn 0.24.1 documentation. url: https://scikit-learn.org/stable/modules/sgd.html

[32] sklearn.svm.LinearSVC — scikit-learn 0.24.1 documentation. url: https ://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.htm