The Purpose of The CICIoT 2023 Dataset and It's Relevance to Cybersecurity

line 1: 1st Abdulkadir Efe Buğday
line 2: *Computer Engineering*
(of Biruni University)
line 3: *Biruni University*
line 4: Istanbul, Turykey
line 5: 210408033@st.biruni.edu.tr

line 1: 2nd Ali Uğur Gümüşlü
line 2: *Computer Engineering*
(of Biruni University)
line 3: *Biruni University*
line 4: Istanbul, Turykey
line 5: 210408007@st.biruni.edu.tr

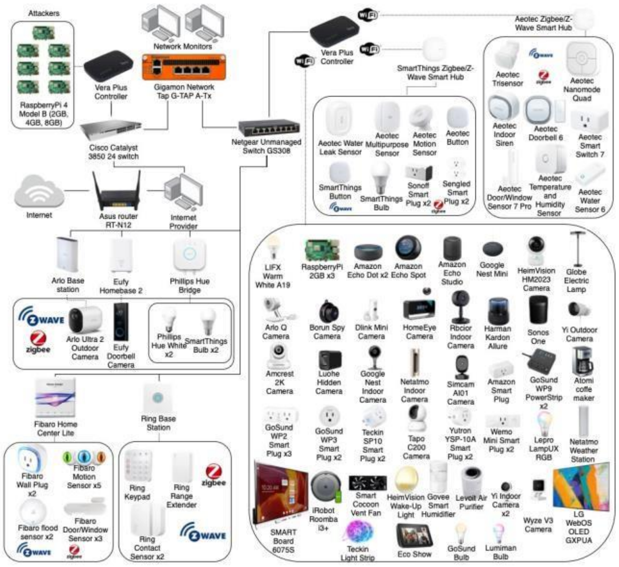
Abstract— The dataset is focused on IoT attacks, designed to support the development of security analytics tools tailored for real-world IoT environments. To create this dataset, 33 distinct types of attacks were conducted within an IoT network consisting of 105 interconnected devices. These attacks were categorized into seven main categories: DDoS, DoS, Recon, web-based attacks, brute force attacks, spoofing and the Mirai botnet. Each attack originated from compromised IoT devices targeting other devices within the same network.[1],[2]

Keywords—IoT, Cybersecurity, Machine Learning, Dataset

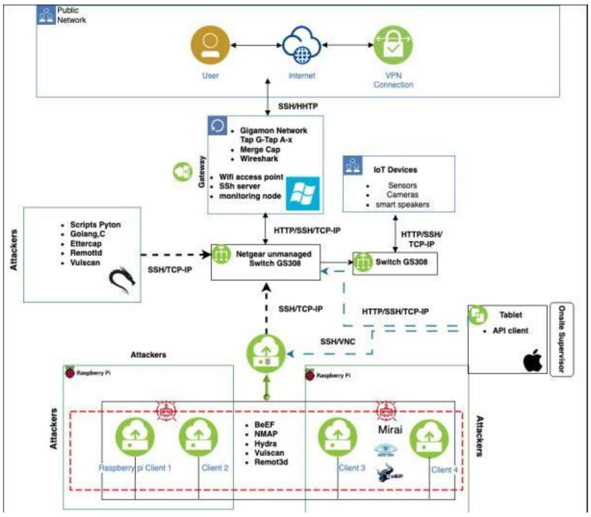
INTRODUCTION

The production of IoT security data that can be used to support real applications is challenging for several reasons. One of the main problems is having an extensive network composed of several real IoT devices, similar to topologies of real IoT applications. Similar works use simulated or very few IoT devices due to the costs of network equipment required (e.g., switches, routers, and network tap), and security analysists who dedicated to maintaining such an infrastructure. Thereupon, the Canadian Institute for Cybersecurity (CIC) has a distinguished presence in the cybersecurity ecosystem and a history of high-impact contributions to industry and academia. This success enabled CIC to establish an IoT lab with a dedicated network to support the development of IoT security solutions. The dataset is especially relevant to the development of intrusion detection systems (IDS) and machine learning models for IoT security. Additionally, it addresses a critical gap in IoT security research: the need for data from malicious IoT devices rather than traditional computing systems as IoT devices have distinct characteristics and vulnerabilities that complicate attack detection and mitigation.[1],[2]

CIC IoT Lab



The IoT topology deployed to produce the CICIoT2023



Attack framework for the dataset

FEATURES IN DETAIL

flow_duration: The total time from the start to the end of a network flow.

Header_Length: The combined length of headers in each packet of the flow.

Protocol Type: The protocol used for the flow. (e.g. TCP, UDP, ICMP…)

Duration: The time duration of the flow or the connection.

Rate: The average rate of packet transmission within the flow.

rate: The rate of packets sent from the source to the destination in the flow.

Drate: The rate of packets sent from the destination to the source in the flow.

fin_flag_number: The number of packets with the FIN flag set that indicates the end of a connection.

syn_flag_number: The number of packets with the SYN flag set. Generally the start of a TCP connection.

rst_flag_number: The number of packets with the RST flag set. Signaling an abrupt connection termination.

psh_flag_number: The number of packets with the PSH (push) flag set. Indicating data should be sent immediately.

ack_flag_number: The number of packets with the ACK (acknowledge) flag set. Confirming received packets.

ece_flag_number: The number of packets with the ECE (Explicit Congestion Notification Echo) flag set indicating congestion.

cwr_flag_number: The number of packets with the CWR (Congestion Window Reduced) flag set. Acknowledging congestion control.

ack_count: Total ACK flags counted in the flow.

syn_count: Total SYN flags in the flow.

fin_count: Total FIN flags in the flow.

urg_count: Total URG (urgent) flags in the flow.

rst_count: Total RST flags in the flow.

HTTP, HTTPS, DNS, Telnet, SMTP, SSH and IRC: The number of packets in the flow using each application layer protocol.

TCP, UDP: The number of packets in the flow using the TCP and UDP protocols.
DHCP, ARP, ICMP: The number of packets in the flow using the DHCP, ARP, and ICMP protocols.

IPv: The count of packets with IP versions used in the flow.

LLC: The count of Logical Link Control layer packets, often used in network layer flows.

Tot sum: The sum of all packet sizes within the flow.

Min, Max: The minimum and maximum packet sizes in the flow.

AVG: The average packet size in the flow.

Std: The standard deviation of packet sizes in the flow (The variation).

Tot size: The total size of the payload within the flow.

IAT: Inter-Arrival Time, the average time gap between packets in the flow.

Number: Total packet count in the flow.

Magnitude: Indicates the magnitude of changes in packet size.

Radius: A feature derived from the spatial distribution of packet transmission.

Covariance: Measures the correlation between packet sizes across the flow.

Variance: The variance in packet size, indicating variability within the flow.

Weight: The total "weight" of the flow calculated by summing various attributes like packet counts, size and duration.

Label: The label associated with the flow, indicating whether it's benign or a specific type of attack (e.g. DDoS, Brute Force, Spoofing) used for supervised learning tasks.

THE CURRENT LITERATURE

The transition from traditional to machine learning-based detection involves the following techniques. Traditional signature-based IDS has difficulties in real-time detection and detection of unknown threats because it's always requires a database update, Neural Networks, Support Vector Machines, Decision Trees and Ensemble Methods will help improve detection by learning patterns in the data to take on new types of threats.

Supervised Methods: Supervised Learning Algorithms like Decision Trees, Support Vector Machines and Neural Networks such as Radial Basis Function show very good performance with labeled data.

Unsupervised Methods: K-Means and Fuzzy C-Means are effective to find unknown anomalies in unlabeled datasets, but may increase the rate of false positives.

Hybrid Models: It combines the supervised and unsupervised methods in order to balance the trade-offs between precision and adaptability.

Model Performance and Evaluation: Dataset quality and the features selected carries high value model performance. Precision, recall, accuracy, ROC, etc. gives a crystal clear overview of the model's efficiency.

Radial Basis Function: Has continuously shown very high precision, recall, and ROC scores, and will continue to be one of the best choices.

Feature Selection and Preprocessing: Features Extracted from Network Traffic, for example, entropy of source and destination IP, types of protocols used, have been highly important in successful detection.
Two feature selection methods feature, CfsSubsetEval and InfoGainAttributeEval, helped improve the model's performance by focusing resources on the most relevant data points of the entire dataset.

Real-World Challenges to Deployment:
1- False Alarm Rates: Too many false positives overwhelm the analyst and undermine trust in a system.
2- Training Complexity: Supervised learning requires a large amount of labeled data in big networks, which is resource-intensive.
3- Operational Adaptability: The ML models should adapt to network behaviors that keep on changing and to new threat vectors.

BEST PRACTICES

1- Multi-Method Coupling for Detection:

- Hybrid model and ensemble techniques does increase the strength of multiple ML approaches.
- Ensemble methods contain robust performance by combining the decisions of multiple algorithms.

2- Advanced Feature Engineering Leverage:

- Statistical measures like entropy and dimensionality reduction techniques like PCA to represent the data in a better manner.
- Utilizing the relevant subsets of features to enhance the accuracy of detection with lower computation load.

3. Contextual Model Deployment:

- Adapting the ML models to specific operational contexts, taking into consideration network size, traffic patterns, and threat landscapes.
- Periodically refresh models as emerging threats demand it to maintain accuracy of the model.

4. Comprehensive Evaluation Metrics Utilized:

- Model performance is assessed on Precision, Recall, Accuracy, and ROC curve to capture the performance of the model for better clarity.
- Thresholds for detection should be optimally set to balance sensitivity and specificity.

5. Continuous Monitoring and Adaptation:

- Adaptive learning systems have been implemented to improve detection capabilities constantly when more data becomes available.
- Integration with traditional security measures, such as firewalls and antivirus anomaly detection, provides layered defense.

6. Effective Training and Data Management:

- The use of real-world datasets has included training on Kyoto 2006+, KDD'99, among others, that ensures the models are tested against real scenarios.
- To simulate attack scenarios and fill the datasets, honeypots would be considered.

7. Decreasing False Alarms:

- Feature selection and fine-tuning are used extensively to keep false alarms at a minimum.
- Validate the results of detection in real life on practicality and reliability.

WHY DATA PREPROCESSING IS IMPORTANT

Data preprocessing provides to deal with missing values, non-numeric values, duplicates, and noise to clear the dataset for reliable data preprocessing.

Also, data preprocessing selects and retains, from among the available feature variables, the most appropriate ones, which makes handling the dataset easier and enhances model performance.

Normalization of data ensures that all variables contribute equally in the analysis, thus avoiding bias due to variation in the range of features.

Preprocessing decreases irrelevant or misleading information, thereby reducing the probability of false alarms or any blind spot which may be used within any detection system.

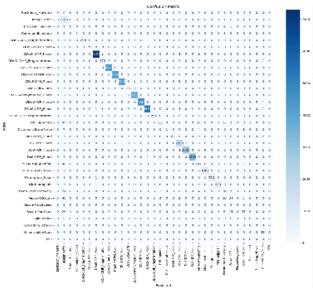


Figure 1.

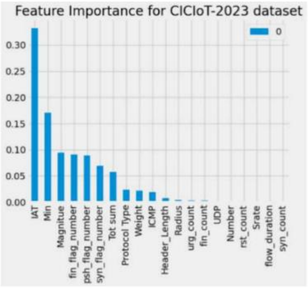
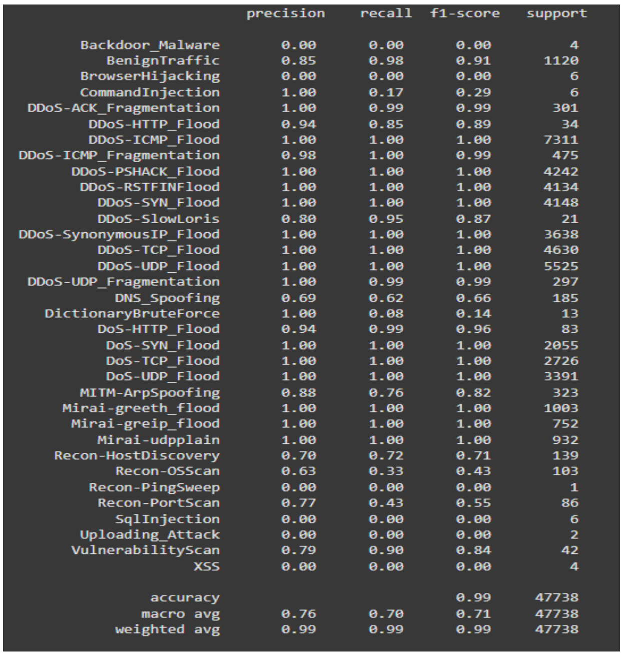


Figure 2.



```
                          precision    recall  f1-score   support

         Backdoor_Malware       0.00      0.00      0.00         4
            BenignTraffic       0.85      0.98      0.91      1120
         BrowserHijacking       0.00      0.00      0.00         6
         CommandInjection       1.00      0.17      0.29         6
   DDoS-ACK_Fragmentation       1.00      0.99      0.99       301
          DDoS-HTTP_Flood       0.94      0.85      0.89        34
          DDoS-ICMP_Flood       1.00      1.00      1.00      7311
  DDoS-ICMP_Fragmentation       0.98      1.00      0.99       475
        DDoS-PSHACK_Flood       1.00      1.00      1.00      4242
         DDoS-RSTFINFlood       1.00      1.00      1.00      4134
           DDoS-SYN_Flood       1.00      1.00      1.00      4148
           DDoS-SlowLoris       0.80      0.95      0.87        21
  DDoS-SynonymousIP_Flood       1.00      1.00      1.00      3638
           DDoS-TCP_Flood       1.00      1.00      1.00      4630
           DDoS-UDP_Flood       1.00      1.00      1.00      5525
   DDoS-UDP_Fragmentation       1.00      0.99      0.99       297
             DNS_Spoofing       0.69      0.62      0.66       185
    DictionaryBruteForce       1.00      0.08      0.14        13
           DoS-HTTP_Flood       0.94      0.99      0.96        83
            DoS-SYN_Flood       1.00      1.00      1.00      2055
            DoS-TCP_Flood       1.00      1.00      1.00      2726
            DoS-UDP_Flood       1.00      1.00      1.00      3391
         MITM-ArpSpoofing       0.88      0.76      0.82       323
       Mirai-greeth_flood       1.00      1.00      1.00      1003
        Mirai-greip_flood       1.00      1.00      1.00       752
          Mirai-udpplain       1.00      1.00      1.00       932
      Recon-HostDiscovery       0.70      0.72      0.71       139
            Recon-OSScan       0.63      0.33      0.43       103
          Recon-PingSweep       0.00      0.00      0.00         1
          Recon-PortScan       0.77      0.43      0.55        86
            SqlInjection       0.00      0.00      0.00         6
         Uploading_Attack       0.00      0.00      0.00         2
       VulnerabilityScan       0.79      0.90      0.84        42
                     XSS       0.00      0.00      0.00         4

                accuracy                           0.99     47738
               macro avg       0.76      0.70      0.71     47738
            weighted avg       0.99      0.99      0.99     47738
```

Figure 3.

Shown in the Figure 3. High precision, recall, and F1-scores for major DDoS attack types suggest the model is very effective for these attack categories.

Lower performance for minority classes (e.g., Recon-PingSweep, SqlInjection, Uploading_Attack) indicates a potential imbalance in the dataset or insufficient training samples for these labels.

Macro Average (precision: 0.76, recall: 0.70): Indicates that minority classes are underperforming.

Weighted Average (precision, recall, F1 close to 0.99): Skewed by the dominance of well-represented classes.

PREFERRED DATA MINING TECHNIQUE

For the dataset we are using, classification is the most suitable technique because:

The CIC IoT Dataset contains a column named as “label” which shows the type of network activity (e.g., DDoS-RSTFINFlood, DoS-TCP_Flood, BenignTraffic). Labeled data means each data point already has a known output (class), this type of columns in datasets makes classification type models such as Random Forest possible. Classification models can learn patterns from this labeled data during training and use these patterns to predict the label for new, unseen data (train).

Using classification, the model learned threat patterns to identify relationships between the features (e.g., packet size, connection type) and the labels (e.g., DoS-TCP_Flood, BenignTraffic). This automation allows cybersecurity systems to respond faster to potential threats, reducing the time to detect and mitigate attacks. If a packet has features similar to those associated with DDoS-RSTFINFlood, the model will classify it as such. If the features align with BenignTraffic, the model will classify it as non-threatening.

TYPE of THREATS

In the cybersecurity field IT employees heavily relies on identifying and categorizing network traffic to detect threats in a short time. Automating the detection and categorization of threats helps in real-time decision-making and faster incident response.

Based on the classification results and the dataset, the types of threats detected and their cybersecurity implications are as follows:

1- Distributed Denial of Service (DDoS) Attacks:

DDoS attacks overwhelm systems by flooding them with excessive requests, rendering services unavailable to legitimate users. They can disrupt businesses, financial services, and critical infrastructure.

2- Malware and Exploits:

Implications: Malware like backdoors allows attackers unauthorized access to systems, enabling data theft or further exploitation. Exploits such as SQL Injection and Command Injection target application vulnerabilities, leading to data breaches or system compromise. These threats have low support in the dataset, resulting in poor detection performance.

3- Reconnaissance Activities:

Reconnaissance activities aim to gather information about systems, networks, or services, often as a precursor to a targeted attack. Identifying these activities early can prevent attackers from escalating their attacks.

4- **Man-In-The-Middle (MITM) Attacks:**

MITM attacks allow attackers to intercept and manipulate communications between two parties, leading to credential theft, data manipulation, or espionage. Detection of MITM attacks is critical in preventing sensitive information from being compromised.

5- **Brute Force and Dictionary Attacks:**

These attacks attempt to guess credentials by systematically trying combinations, potentially leading to unauthorized access. Early detection can prevent attackers from gaining entry to critical systems.

6- **Vulnerability Scanning:**

Vulnerability scanning aims to identify security weaknesses in systems or applications. While this can be part of legitimate security testing, unauthorized scans are often the precursor to attacks.

Classes with low support (e.g., Backdoor_Malware, XSS, Uploading_Attack) show poor performance, indicating that the model struggles to detect rare or underrepresented threats.
This is a critical issue, as these rare threats can be highly impactful if undetected.

SUMMARY

The Random Forest Classifier with the accuracy of 99.47% is highly effective in handling large datasets with many classes, especially when there is a clear distinction among the dominant labels.
The use of StandardScaler ensures numerical stability during training, contributing to the high accuracy. But the model struggles with underrepresented classes, as seen from low scores for classes like Backdoor_Malware, SqlInjection, and Uploading_Attack.
This could be addressed by increasing the representation of minority classes or applying techniques like SMOTE (Synthetic Minority Oversampling Technique) or using a class-weighted loss function.

CONCLUSION

The applied techniques are highly effective for the dominant attack types, but the model's performance for minority classes could be improved with further preprocessing and model adjustments. The current approach is robust for real-world scenarios dominated by frequent attacks but requires fine-tuning to identify rare attack types reliably.

References

[1] https://www.unb.ca/cic/datasets/iotdataset-2023.html - Canadian Institute for Cyber Security

[2] Neto, E.C.P.; Dadkhah, S.; Ferreira, R.; Zohourian, A.; Lu, R.; Ghorbani, A.A. CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment. *Sensors* **2023**, *23*,5941.

https://doi.org/10.3390/s23135941

[3] Jony, A.I. and Arnob, A.K.B., 2024. A long short-term memory based approach for detecting cyber attacks in IoT using CIC-IoT2023 dataset. Journal of Edge Computing [Online], 3(1), pp.28–42. Available from: https://doi.org/10.55056/jec.648

[4] Y. -X. Meng, "The practice on using machine learning for network anomaly intrusion detection," 2011 International Conference on Machine Learning and Cybernetics, Guilin, China, 2011, pp. 576-581, doi: 10.1109/ICMLC.2011.6016798.

[5] M. Zaman and C. -H. Lung, "Evaluation of machine learning techniques for network intrusion detection," NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, Taiwan, 2018, pp. 1-5, doi: 10.1109/NOMS.2018.8406212.

[6] V. S. A. Raju and S. B, "Network Intrusion Detection for IoT-Botnet Attacks Using ML Algorithms," 2023 7th International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), Bangalore, India, 2023, pp. 1-6, doi: 10.1109/CSITSS60515.2023.10334188.