Denial-of Service (DoS) Attack Detection Using Edge Machine Learning

Ngoc Suong Huynh
Department of Computer Science
Texas State University
San Marcos, USA
bez15@txstate.edu

Sebastian De La Cruz
Department of
Electrical and Computer Engineering
Florida International University
Miami, USA
sdela098@fiu.edu

Alexander Perez-Pons
Department of
Electrical and Computer Engineering
Florida International University
Miami, USA
aperezpo@fiu.edu

2023 International Conference on Machine Learning and Applications (ICMLA) | 979-8-3503-4534-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICMLA58977.2023.00264

***Abstract*—Developing lightweight algorithms to implement DoS attack mitigation on edge devices is a growing interest in edge cybersecurity. Various types of micro-controller boards can be programmed to capture network traffic and implement lightweight machine learning models to analyze the supplied traffic data for signs of intrusion and attacks. This study experimented with building Support Vector Machine and Logistic Regression models on real-time DoS attack scenario data and the CICIoT2023 dataset. The main contribution of this study is to propose a framework for data capturing, processing, and analysis to produce edge machine learning models for DoS attack mitigation.**

***Index Terms*—Denial-of-Service, DoS, machine learning, tinyML, micro-controller**

I. Introduction

Denial-of-Service (DoS) attacks are among the most common and malicious types of cybersecurity attacks. These attacks attempt to disrupt the normal traffic and function of a targeted server or network. The current cybersecurity landscape, which involves an exponentially growing number of connected devices, has also contributed to the evolution of Distributed Denial-of-Service (DDoS) attacks. These attacks control networks of internet-connected devices to direct large-scale and powerful attacks on their targets. DoS and DDoS attack mitigation is of particular interest to the industry of internet-of-things (IoT) devices because the growing adoption of IoT devices render IoT networks an ideal amplification platform for conducting dangerous high-volume attacks [3]. DoS attack mitigation has traditionally been implemented on the cloud; however, as the number of IoT devices increases, the efficiency and viability of cloud security infrastructure diminishes due to high computation and traffic load [4]. Lightweight algorithms are a promising solution to this problem. These algorithms are designed to consume fewer computational resources (e.g. memory, processing power, etc.), thus suitable for implementation on battery-powered, portable or remote edge devices, which reduces the burden of centralized data processing compared to cloud infrastructure. In this study, we will investigate lightweight algorithms for implementation on edge devices to address the limitations of cloud computing in mitigating DoS attacks.

DoS attacks can be launched on all seven layers of the OSI model. In this study, we focus on the network layer, within which there are three major categories of DoS attacks: volumetric, amplification, and protocol-related attacks [1].

User Datagram Protocol (UDP) Flood attack is a volumetric attack that takes advantage of UDP protocol vulnerabilities. The attacker sends a large number of IP packets containing UDP packets to arbitrary ports on a targeted server from a spoofed IP address. The victim server will attempt to respond to these UDP packet requests with appropriate ICMP packets. Responding to a very large number of packets prevents the server from processing legitimate traffic [13].

Amplification attacks exploit a disparity in bandwidth cost between the attacker and the targeted server. The attacker sends small requests that result in large responses, thereby exhausting both the inbound and outbound bandwidth of the victim. Some examples of amplification attacks are ICMP floods and Smurf attacks, both of which overwhelm the victim by sending ICMP Echo requests with IP address spoofed as that of the targeted server. ICMP flood attacks use botnets with spoofed IP addresses to increase the volume of the attack and make it difficult to detect the source. On the other hand, Smurf attacks exploit immediate IP broadcast networks to transmit fake Echo requests to all hosts on the network of targeted server, triggering them to send back an overwhelming number of responses [1].

SYN flood attacks exploit the three-way handshake required in a Transmission Control Protocol (TCP) connection. The attacker sends a high volume of SYN packets to the targeted server, but never responses to the server's SYN/ACK packets. This process triggers the server to leave ports open to receive the ACK response from the attacker, exhausting the number of ports available for legitimate requests [1].

II. Related Works

A. Early Detection Using Edge Devices

C. Avasalcai et al. [4] identified several limitations of cloud computing with regard to the current state of the art in IoT: (1) high end-to-end (e2e) latency, (2) increased risk of congestion and bandwidth waste, and (3) difficulties of satisfying current data privacy and other requirements of new IoT applications.

1946-0759/23/$31.00 ©2023 IEEE
DOI 10.1109/ICMLA58977.2023.00264

They proposed shifting to edge paradigms to enable lower e2e latency, more responsive IoT applications, augmented scalability and privacy with processing data at the edge. These benefits of edge computing are particularly relevant to cybersecurity and DoS attack mitigation. Z. Liu et al. [3] highlighted three main characteristics of DDoS attacks in edge environment: (1) hard to detect due to indiscernible flow classified features from normal traffic, (2) low-cost and able to attack multiple targets simultaneously, (3) long term attacked-targets become insensitive to attacks. For these reasons, edge computing is becoming an essential solution to "alleviate the traffic load of the network," and at the same time "reduce the delay of defense decision and improve the response speed." Z. Liu et al. suggested an edge DDoS detection method with high accuracy, fast response time and certain self-learning ability for unknown new attacks using LR and Deep CNN Q-Network models [3].

The cumulative advantages of edge detection of anomalous network traffic promote early and prompt attacks mitigation, preventing such attacks from spreading within the network and causing further network failure and property loss. Moreover, these advantages allow IoT devices and networks to satisfy increasingly strict requirements regarding latency, data privacy, and scalability.

B. Machine Learning Models for DoS Attack Detection

Current DDoS detection methods are often based on analyzing patterns in network packet metadata to discern anomalous and normal network traffic. Therefore, machine learning models have been widely implemented in cybersecurity and DDoS detection for their quality performance while fulfilling complex data analysis tasks.

In particular, M. F. Ashfaq et al. [5] suggests that machine learning techniques can assist in restricting false positives. Moreover, neural network models, a subset of machine learning models, can overcome classical machine learning models' limitations in handling large datasets. They analyzed two datasets: one collected using Wireshark and the other imported from the KDD-Cup database . Two machine learning models were used to train these datasets: Logistic Regression (LR) and Decision Tree (DT), both of which outperformed K-nearest neighbors (K-NN) and support vector machine (SVM) in terms of accuracy across experiments on different numbers of features . While K-NN and SVM models achieved accuracy ranging from 90% - 98% for binary and 10-feature sets, the figures for LR and DT models were between 99.81% and 99.89% [5].

Y. Jia et al. [6] applied long short-term memory (LSTM) and convolutional neural network (CNN) models trained by the CICDDoS2019 dataset [14] to the data generated in a real attack scenario in a computer-based edge server. CICDDoS2019 contains both benign and 12 most common types of DDoS attacks launched from PCs. After attempting a number of layers for both LSTM and CNN from 1 to 10, they discovered that 3 layers for LSTM and 6 layers for CNN are the most effective ones. The average accuracy results for LSTM and CNN models are up to 98.9% and 99.9%, respectively.

C. Benchmark Dataset

The characteristics of IoT devices pose a challenge against detecting and mitigating cybersecurity attacks, prompting demand for robust datasets to train efficient DDoS detection models. Most existing datasets such as the CICDDoS2019 dataset do not consider relatively uncommon or emerging types of attacks. In addition, they only consider computer-based attacks rather than those launched from malicious IoT devices. To fill this gap, E.C.P. Neto et al. [2] introduces the CICIoT2023 dataset, composed of 33 attacks simulated on 105 devices. They used five machine and deep learning algorithms to evaluate this dataset: Logistic Regression, Perceptron, AdaBoost, Deep Neural Network (DNN), and Random Forest (RF). Results showed that all five models maintain high performance in binary classification. In 8-class and 34-class classifications, DNN, LR and Decision Tree models maintain relatively high performance, especially in terms of accuracy.

D. Tiny Machine Learning (TinyML)

As powerful a tool as machine learning is, many challenges evolve in implementing machine learning in edge devices. Training machine learning models is computationally and time-wise expensive, and the power and memory consumption for which often exceeds the constraints of edge devices [7]. TinyML, an emerging paradigm on the intersection between machine learning and low-power computing, solves the above challenges by enabling the implementation of machine learning algorithms on ultra-low-power devices. Ultra-low-power devices are those designed to operate and process data with the smallest amount of power needed, typically under a milliWatt [8]. This study primarily focuses on ultra-low-power microcontroller units (MCUs), which typically have an SRAM (static random access memory) between 256KB and 1MB, and clock rates between 64MHz to 600MHz [8].

There are a wide variety of MCUs that allow for TinyML deployment. B. Sudharsan et al. [8] compared the costs and performance of seven different common boards and showed that between the two most inexpensive boards, Raspberry Pi Pico and ESP32, the latter had the best price-performance. This makes ESP32 a promising option for producing high-performance yet affordable smart devices.

So far, TinyML has been implemented dominantly in keyword spotting, image classification and visual wake words. Less common use cases include object detection, anomaly detection, motor control, gesture recognition, and face recognition [9].

III. Proposed Work

S. S. Saha et al. [11] introduced a coherent and closed loop machine learning model development and deployment workflow for micro-controllers and discussed the current prominent model development software suites. The process consists of the model development phase and the model deployment phase. The model development phase involves using data engineering frameworks to collect and clean raw data to produce a dataset; optionally, feature projection can also be performed

at this stage to reduce dimensionality (i.e reduce the number of features analyzed while retaining the meaningful properties of the original data [15]). Several models are then chosen from a pool of "lightweight model zoo" based on the application and hardware constraints. The model deployment phase involves porting the best performing model to a TinyML software suite, performing model compression along generating embedded code, then flashing the C file system onto the micro-controller for inference [11].

In this study, we aim to train machine learning models on network traffic data captured during a DoS attack scenario using a packet sniffer on micro-controller. The performance of these models will be examined further using the CICIoT2023 dataset, which contains a wide range of the most common and recent types of DoS attack. As we focus on implementing our project on Espressif ESP32 micro-controllers, we will use TinyML frameworks compatible with ESP32 boards such as TensorFlow Lite Micro to compress, convert and deploy these models onto an ESP32 board. Once the models are deployed, they should enable the micro-controller to perform inference when supplied with real-time traffic data.

IV. Implementation

A. Packet Sniffing Using Micro-controllers for Machine Learning

Packet sniffing is a method of intercepting and inspecting each packet flowing across a network. Packets sniffers are programs which are used to read packets that travel across the network layer of the Transmission Control Protocol/Internet Protocol, by listening to data that arrives at the Network Interface Card (NIC). Packet sniffers exist in Local Area Networks (LANs) and Wide Area Networks (WANs) [10].

In this study, we built and deployed a packet sniffer on ESP32 micro-controllers to enable them to monitor and detect malicious network traffic. The steps used for creating a Linux packet sniffer can also be implemented on micro-controllers using the appropriate development framework or environment. The main steps in the development of a packet sniffer are: (1) creating a socket stream, (2) setting the capturing interface into promiscuous mode, so that it accepts all packets in the network, and (3) reading data from the open socket stream [10]. We also ran Wireshark in tandem with packet sniffing to capture data for machine learning model development.

B. Data Capturing

Our data capturing setup consists of a computer which acts as a server and several computers as clients. The client computers are connected to the server wirelessly via a TP-Link 1200 Dual-Band wireless access point. We launched a De-authentication DoS attack from one of the client computers in the network and captured all network traffic using an ESP32 Pico-Kit board flashed with a packet sniffer set in promiscuous and monitor mode. We connected the ESP32 Pico-Kit board to one of the client computers and captured traffic data into PCAP files using Wireshark.

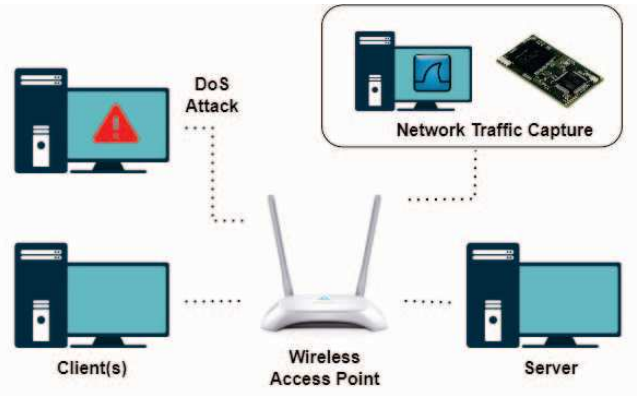

Fig. 1. DoS Attack Network Traffic Data Capture Setup

C. Machine Learning Model Development

Real-time network traffic data captured using Wireshark can be parsed with DPKT module in Python or CICFlowMeter traffic flow analyzer to extract features on packet metadata and packet flow statistics [2]. In this study, we extracted two features from captured network traffic data: (1) frame length, and (2) packet inter-arrival time. We used the preprocessed dataset to train Support Vector Machine (SVM) and Logistic Regression (LR) models.

Both SVM and LR are relatively lightweight machine learning models that are robust for datasets with fewer features. SVM is more lightweight than most models in terms of training time, model size, inference time and ease of implementation, especially on constrained environments such as micro-controllers. SVM models have also proven robust when analyzing larger numbers of features (up to 34) as experimented in study [12]. LR models are easy to implement for binary as well as multi-classification problems to account for different types of DDoS attacks at the same time. Compared to other models, it is also less prone to overfitting [5]. These models fit our purpose of experimenting viable machine learning models with minimal memory and computation footprint.

Besides real-time captured data, we also used DoS attack data from the CICIoT2023 dataset to train the aforementioned models. This dataset contains more types of DoS attacks experimented on a wide range of IoT devices. As we only experimented with a simplified Deauthentication DoS attack on a small Wi-Fi network, the CICIoT2023 dataset assists in testing the applicability and generalizability of our models over the most common and recent types of DoS attacks. The features that we used for analyzing the CICIoT2023 Dataset are flow duration, frame length, protocol type, time to live, flow rate, total length of frames in flow, and inter-arrival time (IAT).

V. Results and Analysis

We evaluated our machine learning models based on two metrics: accuracy and confusion matrix. Both datasets are used

to train binary classification models (benign against malicious traffic) using SVM and LR models.

Table I shows the precision, recall and F1-score in detecting malicious traffic, and overall accuracy for four models.

TABLE I
CLASSIFICATION REPORT

		SVM	LR
CICIoT2023	Accuracy	99.36%	99.15%
	Precision	99.44%	99.18%
	Recall	99.85%	99.89%
	F-1 Score	99.65%	99.53%
Real-time	Accuracy	99.84%	99.71%
	Precision	100.00%	100.00%
	Recall	99.69%	99.13%
	F-1 Score	99.84%	99.56%

Both LR and SVM models yielded comparable results in terms of accuracy, precision, recall and F-score, with figures for models trained on the CICIoT2023 dataset slightly lower than those of models trained on real-time data. The results can be further examined by analyzing their confusion matrices.

Fig. 2 and Fig. 3 show the confusions matrices for SVM and LR models trained on real-time data. Malicious instances are labeled '1' and benign instances '0'. Both SVM and LR models yielded high true positive rates of 99.69% and 99.44%, respectively, indicating they correctly classified almost all DoS attack instances as malicious. The percentage of false positives were 0.31% for SVM and 0.56% for LR, while neither models returned any false negatives. While both models achieved high accuracy, precision and recall, they were biased towards classifying traffic as malicious.

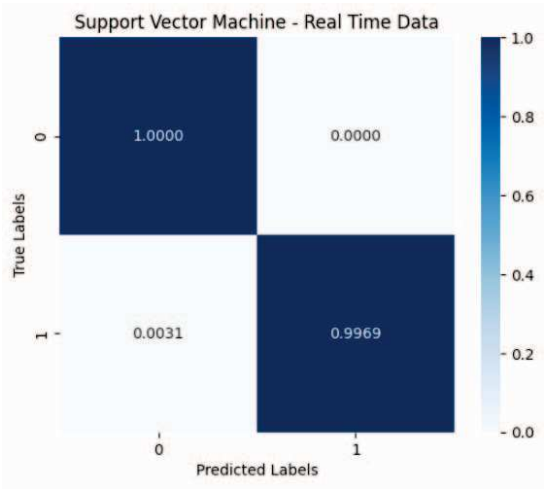

Fig. 2. Confusion matrix for SVM trained on real-time data

Fig. 4 and Fig. 5 show the confusion matrices for SVM and LR models trained on the CICIoT2023 dataset. Compared to models trained on real-time data, these models reported some percentages of both false positives and false negatives. While the percentages of false negatives were relatively low (0.11% for LR and 0.15% for SVM), both models yielded significantly higher rates of false positives (5.04% for SVM and 6.20% for LR). As our real-time data only contains instances of De-authentication DoS attacks while the CICIoT2023 dataset includes a much wider variety of DoS and DDoS attacks, classifying network traffic proves more challenging for models trained the CICIoT2023 dataset. While these models achieved overall lower accuracy and precision, they are expected to correctly detect many types of DoS attacks in real-life scenarios.

Fig. 3. Confusion matrix for LR trained on real-time data

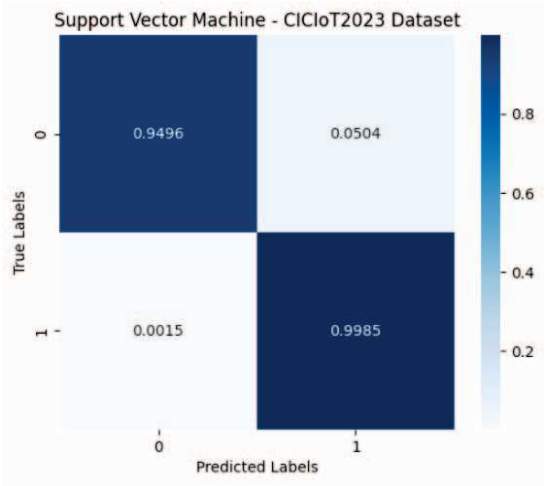

Fig. 4. Confusion matrix for SVM trained on CICIoT2023 Dataset

We observed that compared to models trained on CICIoT2023 dataset, models trained on real-time data yielded higher accuracy, but were more prone to poor generalization to other data due to (1) real-time experiments involving a single type of DoS attack and (2) the limited number of features available for analysis. This limitation can be improved by incorporating a variety of DoS attacks and implementing more thorough feature extraction processes in future experiments.

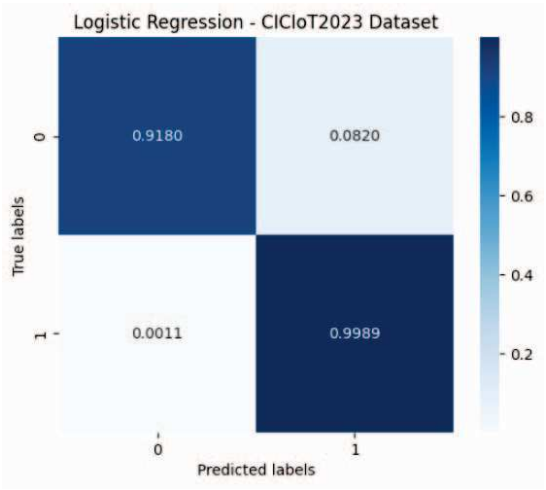

Fig. 5. Confusion matrix for LR trained on CICIoT2023 Dataset

VI. Conclusion and Future Work

Implementing DoS attack detection and mitigation on the edge will alleviate the traffic load within IoT networks, lower end-to-end latency, improve response speed and improve data privacy. A survey of current micro-controllers and software development frameworks suggests the feasibility of capturing network traffic data and deploying lightweight machine learning models on small edge devices.

This idea still needs to be tested on a variety of micro-controller boards to study their capability to capture network traffic and perform inference on the resulting data. In the future, we aim to investigate more novel and advanced micro-controller boards and experiment with their capability to capture various types of network traffic and execute DoS attack mitigation decisions based on lightweight yet robust machine learning models. Besides classical machine learning models, we aim to experiment with deep neural network models in the future to study the trade-offs between memory, computation load, and performance across a wide range of machine learning models.

References

[1] R. S. Devi, R. Bharathi and P. K. Kumar, "Investigation on Efficient Machine Learning Algorithm for DDoS Attack Detection," *2023 International Conference on Computer, Electrical & Communication Engineering (ICCECE)*, Kolkata, India, 2023, pp. 1-5, doi: 10.1109/ICCECE51049.2023.10085248.

[2] E.C.P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, A.A. Ghorbani, "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," *Sensors*, 2023, pp. 23, doi: 10.3390/s23135941.

[3] Z. Liu, X. Yin and Y. Hu, "CPSS LR-DDoS Detection and Defense in Edge Computing Utilizing DCNN Q-Learning," in *IEEE Access*, vol. 8, pp. 42120-42130, 2020, doi: 10.1109/ACCESS.2020.2976706.

[4] C. Avasalcai, C. Tsigkanos and S. Dustdar, "Decentralized Resource Auctioning for Latency-Sensitive Edge Computing," *2019 IEEE International Conference on Edge Computing (EDGE)*, Milan, Italy, 2019, pp. 72-76, doi: 10.1109/EDGE.2019.00027.

[5] M. F. Ashfaq, M. Malik, U. Fatima and M. K. Shahzad, "Classification of IoT based DDoS Attack using Machine Learning Techniques," *2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, Seoul, Korea, Republic of, 2022, pp. 1-6, doi: 10.1109/IMCOM53663.2022.9721740.

[6] Y. Jia, F. Zhong, A. Alrawais, B. Gong and X. Cheng, "FlowGuard: An Intelligent Edge Defense Mechanism Against IoT DDoS Attacks," in *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9552-9562, Oct. 2020, doi: 10.1109/JIOT.2020.2993782.

[7] N.N. Alajlan, D.M. Ibrahim, "TinyML: Enabling of Inference Deep Learning Models on Ultra-Low-Power IoT Edge Devices for AI Applications," *Micromachines 2022*, pp. 13, 851. doi: 10.3390/mi13060851

[8] B. Sudharsan et al., "TinyML Benchmark: Executing Fully Connected Neural Networks on Commodity Microcontrollers," *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, New Orleans, LA, USA, 2021, pp. 883-884, doi: 10.1109/WF-IoT51360.2021.9595024.

[9] H. Han and J. Siebert, "TinyML: A Systematic Review and Synthesis of Existing Research," *2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Jeju Island, Korea, Republic of, 2022, pp. 269-274, doi: 10.1109/ICAIIC54071.2022.9722636.

[10] S. Ansari, S. G. Rajeev and H. S. Chandrashekar, "Packet sniffing: a brief introduction," in *IEEE Potentials*, vol. 21, no. 5, pp. 17-19, Dec. 2002-Jan. 2003, doi: 10.1109/MP.2002.1166620.

[11] S. S. Saha, S. S. Sandha and M. Srivastava, "Machine Learning for Microcontroller-Class Hardware: A Review," in *IEEE Sensors Journal*, vol. 22, no. 22, pp. 21362-21390, 15 Nov.15, 2022, doi: 10.1109/JSEN.2022.3210773.

[12] S. Allagi, R. Rachh and B. Anami, "A Robust Support Vector Machine Based Auto-Encoder for DoS Attacks Identification in Computer Networks," *2021 International Conference on Intelligent Technologies (CONIT)*, Hubli, India, 2021, pp. 1-6, doi: 10.1109/CONIT51480.2021.9498284.

[13] S. S. Kolahi, K. Treseangrat and B. Sarrafpour, "Analysis of UDP DDoS flood cyber attack and defense mechanisms on Web Server with Linux Ubuntu 13," *2015 International Conference on Communications, Signal Processing, and their Applications (ICCSPA'15)*, Sharjah, United Arab Emirates, 2015, pp. 1-5, doi: 10.1109/ICCSPA.2015.7081286.

[14] I. Sharafaldin, A. H. Lashkari, S. Hakak and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," *2019 International Carnahan Conference on Security Technology (ICCST)*, Chennai, India, 2019, pp. 1-8, doi: 10.1109/CCST.2019.8888419.

[15] N. Ahmad and A. B. Nassif, "Dimensionality Reduction: Challenges and Solutions," *ITM Web of Conferences*, 2020, doi: 10.1051/itmconf/20224301017.