CS 658 Project Report
Group No: 8
Group Members:

SAUGAT KANNOJIA SANKALP PANDE
ROHIT KUMAR MANASVI JAIN

Classification of DDoS Attacks using various Deep Learning Techniques based on CICEV2023 Dataset

Project Link: https://github.com/friedrice123/CS658

Problem Statement:

With the growth of Electric Vehicle (EV) charging infrastructure, Distributed Denial of Service (DDoS) attacks targeting EV authentication systems pose significant threats, potentially disrupting service availability. Data parameters related to EV authentication must be sequentially processed in Charging Stations (CS) . As EVs authenticate one by one, any interruption by another vehicle is restricted. This dependency creates a critical vulnerability: if an attacker triggers continuous authentication failures via DoS or DDoS attacks, EVs may be unable to access charging services. Such attacks can significantly hinder the stable operation of authentication service, posing a serious concern for the overall reliability of EV charging networks.

Most of the latest studies aside from CICEV2023 dataset focuses on detection models for DoS or DDoS attacks have been applied within general networks, lacking focus on the specific challenges in EV charging infrastructure. Current datasets often only capture packet reception counts over a set period, which limits the effectiveness of detection models tailored for EV networks. Therefore, it is essential to develop a detection classifier model, trained on the CICEV2023 dataset using diverse machine learning features, that accurately detects and mitigates DDoS attacks in EV charging networks, ultimately enhancing the resilience and operational stability of these critical systems.

Dataset:

This work is based on a specific dataset, **CICEV2023**, which provides data collected during Distributed Denial of Service (DDoS) attacks against EV charging stations as well as during normal operational states. The dataset includes rich and diverse information capturing the behavior and activities of multiple EV charging stations, covering a wide range of parameters that reflect both attack and non-attack scenarios within the charging infrastructure.

Here, we are using 4 different types of datasets:

- **Correct_ID :** The correct id attack scenario data belongs in this directory.
- **Wrong_CS_TS :** the wrong timestamp data on the charging station belongs in this directory.
- **Wrong_EV_TS :** the wrong timestamp data on the EVs belongs in this directory.
- **Wrong_ID :** the wrong id data of the EVs belongs in this directory.

For each of the above we are subdividing the data based on the randomness and Gaussian attack scenario:

- **Random_CS_Off :** The full attack mode attacks many identical EVs to all CSs simultaneously
- **Random_CS_On :** The random attack mode arbitrarily chooses the victim CS under the attacks.
- **Gaussian_Off :** the data without the Gaussian attack strategy belongs in this directory.
- **Guassian_On :** the data with the Gaussian analysis to create a distribution similar to the normal EV authentication distribution.

For each of the above sub-categories, we have 3 types of json files:

- **STAT.json :** It includes the data points for the attacks and normal scenarios of branch, cycles and instructions for 3 different charging stations
- **TIME_DELTA.json :** It includes the time differences for the attacks and normal scenarios of branch, cycles and instructions for 3 different charging stations
- **TOP.json :** It stores the statistics of various processes like mutex_lock, scheduler, calc_timer_values etc. for attacks and normal scenarios of branch, cycles and instructions for 3 different charging stations.

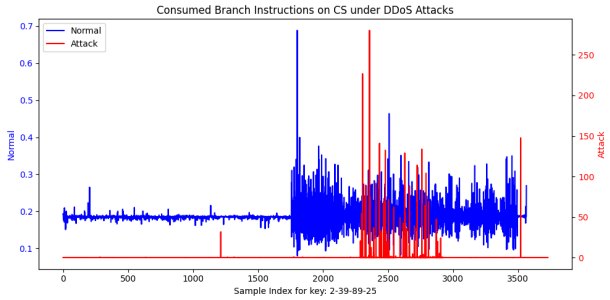
Dataset Generation:

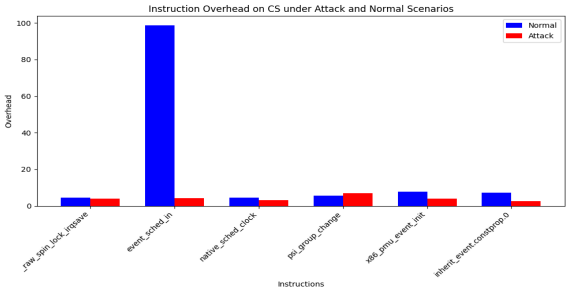
1. The Dataset provided by the authors were given in two formats: Raw and Processed. The Raw Data had txt files of Linux Kernel Overheads, System Performance Data which were proving to be hard to process and extract features. So, the Processed Dataset was used by our group to take all the data that was provided.
2. The directory structure was very complex since there were 4 attack scenarios for which every scenario consisted of 4 combinations of Random/Full Attack and Gaussian/Non-Gaussian based attacks. Each such combination had three types of JSON files for Charging Stations for TOP, STAT and TIME_DELTA data respectively.
3. TIME_DELTA JSON file consisted of nesting based on Charging Station, then further on attack/normal situation which yielded an array of data points observed during the analysis done by the authors. This array of data could not be processed very easily by taking them as features, so we chose to carry out a statistical analysis of the data and converted this data into 4 statistical features i.e. the mean, maximum, minimum and standard deviation observed for each scenario thus generating a CSV file for this named TD.csv.
4. STAT JSON was processed in a similar manner as the TIME_DELTA files and converted into a CSV file named Stat.csv.
5. TOP JSON was tricky to deal with since some metrics observed in attack situations were not present in the normal data, so data cleaning was done such that the metric exists in both attack and normal scenarios so that it can be compared properly. This data was converted taking the Charging Station(CS) data in account, generating a CSV file named Top_CS.csv.

6. After further analysis, we found out that we can create features using all three of these CSV files for which we did a Left Join on all three data thus creating the final CSV file named Combined.csv.

Analysis:

The analysis focuses on understanding the unique characteristics of DDoS attacks targeting EV charging stations. A count analysis was conducted to assess the balance across different classes in the dataset, as well as distributions in the stat and top features, to ensure that the training data provided a representative sample of normal and attack scenarios. Additionally, we analyzed a scatter plot for 12 types of features. We ensured that data was evenly split between attack and normal scenarios to maintain entropy and enable a balanced analysis. Attack patterns were further examined in terms of overhead for branch, cycles, and instructions, offering insights into how DDoS attacks specifically disrupt EV charging services by interfering with sequential authentication processes. This foundational analysis informed the selection of deep learning models capable of processing high-dimensional, sequential data.





Preprocessing:

Data preprocessing involved several steps to enhance model accuracy and reduce overfitting.

1. TOP data had some NULL values which were removed.
2. Standard Scaling was done on the final features to get more robust features.
3. The categorical variables for the Attack/Normal condition were encoded and the unnecessary columns like metric used, ID of the Charging Station and type of metric, were dropped since they didn't provide any relevant information.
4. TOP Data had imbalance in the number of metrics used for Attack scenario which was far greater than the number of metrics used for Normal scenario, so only those metrics were chosen which were present in both of the scenarios.

This preprocessing pipeline contributed to the robustness of the DDoS detection models and was critical in enhancing model reliability.

Feature Type Description:

Feature Name	Description
mean_time_diff	The average interval between consecutive Linux kernel overhead metric measurements (e.g., branch instructions or code symbols) during attack and normal scenarios. This

	value reflects changes in kernel performance and resource usage under different conditions.
max_time_diff	The maximum interval between consecutive Linux kernel overhead metric measurements during attack and normal scenarios, indicating the longest delay observed in kernel performance data collection under different conditions.
min_time_diff	The minimum interval between consecutive Linux kernel overhead metric measurements, indicating the shortest delay in data collection and suggesting periods of rapid metric capture.
std_time_diff	The standard deviation of time intervals between consecutive Linux kernel overhead metric measurements, indicating variability in metric capture frequency and highlighting potential irregularities in kernel performance timing.
mean_stat	Establishes an average behavior baseline of linux kernel overhead metrics, which can be compared with anomalous behavior seen during DDoS attacks.
max_stat	The maximum observed value of linux kernel overhead metrics to detect outliers that might signify unusual or malicious activity.
min_stat	The minimum observed value of a linux kernel overhead metrics, for identifying baseline activity levels in contrast with extremes seen in attack scenarios.
std_stat	The standard Deviation of the linux kernel overhead metrics, capturing the variability in that metrics,to indicate instability or a response to external attack pressures.
mean_top	The average value of linux kernel overhead metrics ,to get get general characteristics of traffic paths or structure in the network during normal operations.
max_top	The maximum value of the linux kernel overhead metrics, potentially indicating unusual traffic routes or extended network paths during periods of high network activity.
min_top	The minimum observed value of the linux kernel overhead metrics, helps to suggest a deviation from standard routing or network topology, possibly due to routing changes during a DDoS event.
std_top	The standard deviation of the linux kernel metrics, showing variability in network routes or structure in response to increased traffic loads.

Methods:

1. A series of deep learning models were developed to accurately detect DDoS attacks. A fully connected neural network was chosen for its flexibility and generalization capacity, and was optimized for performance across several splits of data using cross-validation.
2. 5-fold Cross-validation with 5 random splits was employed initially to ensure consistent model performance across varying data partitions. Key metrics like F1 score and accuracy were averaged across these splits to provide a stable benchmark.
3. Following the cross-validation phase ensuring the validity of the model, a single model was further tuned by adjusting hyperparameters and optimizing for both F1 score and accuracy, with the aim of maximizing detection sensitivity and minimizing false positives.
4. The network architecture consisted of an input layer, two hidden layers with ReLU activation, and a final sigmoid output layer for binary classification.
5. Dropout layers were used to prevent overfitting. Adam and SGD optimizers can be successful under appropriate conditions. For training, the Adam optimiser was used as it was found to be more effective in general situtions, enhancing the convergence of the model.
6. The plot between the training and validation loss was also plotted which showed a decrease in both, suggesting that the data has not gone through overfitting.
7. Finally, the model was evaluated on a separate test dataset to validate its generalization capabilities, with its performance compared against metrics observed during cross-validation and tuning.

Innovation and Comparison with Baseline:

1. The baseline had used a Feed-Forward Network(FFN) along with SGD optimiser with a learning rate of 0.001 which gives an accuracy of 97.6 % .
2. However, the baseline model that was used in the paper we followed for understanding the data used just a single combination from the Random/Full Attack and Gaussian/Non-Gaussian combinations choosing Full and Non-Gaussian based attacks as the sole judge of whether the scenario represents an attack situation which seemed less general to us. So we decided to take all the combinations of the data to make a more general classifier than the baseline model provided.
3. The baseline model uses the extra metrics in the attack scenario rather than creating a balanced dataset as per the paper, which we have resolved by creating balanced data.
4. Using the TIME_DELTA data along with the TOP and STAT data gave a perfect score of 1 for classification without overfitting which led us to believe that it acts almost like a Rule-based classifier, however no such general rules can be made for this.

Results:

The final model achieved full accuracy on the test set, with a high F1-score, indicating balanced precision and recall suggesting that the usage of TIME_DELTA Component almost accurately determines whether an attack is being done or not. This performance underscores the model's capability to accurately detect DDoS attacks while minimizing false positives, which is critical for practical deployment in EV charging infrastructure.

We have used 3 types of features to train the models.

1. The first analysis was done on TOP data since it had the most number of unique features to compare the attack and normals scenarios. However, this data alone was not enough to make an accurate conclusion.
2. To surpass that, STAT data was combine with the TOP data to increase the number of features which slightly increased the performance in classification.
3. When the TIME_DELTA data was also joined with this, the model started to give 100% accuracy almost behaving like a Rule-based classifier.

Results of our models:

Model	Data	Accuracy	Loss	F1-score	Precision	Recall
0	BASELINE	0.976	0.084	0.978	0.971	0.984
1	TOP	80.04±2.01	0.4652	0.79	0.84	0.84
2	TOP+STAT	92.56±1.0	0.1863	0.83	0.94	0.88
3	TOP+STAT+TIME_DELTA	100±0.0	0.0000	1.00	1.00	1.00

Limitations:

1. We were not able to generate more data since the data provided to us was generated using PERF on a particular type of system setting which could not be replicated by us.
2. We could not find any other existing datasets that could add to the current dataset chosen by us. The data provided to us was also limited and didn’t provide a large enough collection to make a more general classifier.
3. The Processed data was generated using some unknown techniques which were not provided to us, thus limiting us with the type of data extracted.
4. This method cannot be used for realtime setting since the data is analysed after extracting it from the system.
5. Any new types of attack scenarios cannot be handled by our proposed model since it is heavily trained on this data.

References:

- https://www.unb.ca/cic/datasets/cicev2023.html
- DDoS detection in electric vehicle charging stations: A deep learning perspective via CICEV2023 dataset