Cybersecurity: Robustness in Incremental Learning

Xiaofeng Zhang, Artificial Intelligence, 0001109513

Abstract

In dynamic cybersecurity environments, machine learning models must adapt to evolving data streams while retaining their ability to detect known threats. This study addresses the dual challenges of adversarial attacks and catastrophic forgetting in class incremental learning systems. Using the **CIC-DDoS2019** dataset, we implemented a neural network-based framework to detect cyberattacks and simulated data poisoning strategies to evaluate their impact. Our experiments revealed how poisoning rates affect model performance, particularly through catastrophic forgetting, where models lose previously learned knowledge. Replay mechanisms were proposed and assessed as a mitigation strategy, showing effectiveness at low to moderate poisoning rates. The findings highlight vulnerabilities in continuous learning systems and provide insights for designing robust, adaptive models for cybersecurity.

1 Introduction

Machine learning is crucial for real-time threat detection in cybersecurity. However, class incremental learning systems, which adapt to new data streams, face two key challenges: adversarial attacks and catastrophic forgetting. Adversarial attacks manipulate input data to compromise detection capabilities, while catastrophic forgetting causes the model to lose previously learned knowledge when exposed to new information. These issues are particularly critical in dynamic environments like IoT networks, where continuous learning is essential.

1.1 Research Objectives

This study aims to:

- Implement a class incremental learning system for detecting cyberattacks using the **CIC-DDoS2019** dataset[1].
- Evaluate the impact of data poisoning attacks on detection performance.
- Analyze catastrophic forgetting as an attack vector and explore replay mechanisms to mitigate its effects.

1.2 Problem Statement

Class incremental learning systems are vulnerable to:

- **Catastrophic forgetting:** Loss of prior knowledge due to new data.
- **Adversarial manipulation:** Exploitation of poisoned data to degrade performance.

These challenges reduce reliability and expose systems to new threats.

1.3 Importance and Contributions

This research highlights the interplay between catastrophic forgetting and adversarial attacks, offering insights into their impact on machine learning systems in cybersecurity. Key contributions include:

- Developing an incremental learning system for cyberattack detection.
- Simulating and analyzing data poisoning strategies.
- Proposing replay mechanisms as a mitigation strategy.
- Providing empirical evidence on robust learning through extensive experiments.

This work provides a foundation for improving the robustness and adaptability of machine learning systems in real-world cybersecurity applications.

2 Dataset

The **CIC-DDoS2019 dataset** is a comprehensive benchmark dataset widely used in evaluating machine learning models for Distributed Denial-of-Service (DDoS) attack detection. This dataset captures various network traffic behaviors under benign and attack scenarios. It comprises eight CSV files, each representing data collected during different working hours, and contains 78 features for each network flow along with a `Label` column indicating whether the flow is benign or represents a specific attack type.

2.1 Data Processing

The raw dataset required several preprocessing steps to ensure its quality and usability for machine learning tasks. Below, we outline each step in the processing pipeline and describe its purpose, methodology, and resulting impact on the data.

2.1.1 Data Loading and Integration

The dataset is stored in eight CSV files, each corresponding to different time periods and scenarios. These files were loaded and concatenated into a single unified dataset to provide a holistic representation of network traffic patterns. This integration resulted in a dataset with a total of **2,826,876 rows and 79 columns**, including 78 feature columns and one label column.

2.1.2 Data Cleaning

Cleaning operations were performed to address missing or irrelevant data:

- **Removal of Irrelevant Columns:** Columns containing only missing values (`NaN`) across all rows were removed, as they provided no useful information for analysis.
- **Handling Missing Values:** Rows with any missing values were dropped to ensure data completeness and integrity. This step ensures that all features contribute to the model training without introducing noise.
- **Standardization of Column Names:** Leading and trailing whitespaces in column names were removed to prevent discrepancies during feature selection or processing.

After cleaning, the dataset was reduced to **2,826,876 rows and 78 feature columns**, along with the `Label` column.

2.1.3 Feature and Label Separation

The dataset was split into:

- **Features:** The 78 columns representing numerical and categorical characteristics of network flows.
- **Labels:** The `Label` column, indicating whether a flow is benign or belongs to a specific attack category.

This separation allows the features to undergo further preprocessing independently of the labels.

2.1.4 Handling Infinite Values

Some numerical features contained infinite values (`inf` or `-inf`), likely arising from division operations in the raw data collection process. These values were replaced with missing indicators and subsequently removed. This ensures numerical stability during model training and prevents issues related to gradient calculations.

2.1.5 Label Encoding

The `Label` column, initially containing categorical values such as 'Benign" or specific attack types (e.g., DDoS"), was encoded into numerical format using label encoding(e.g., "Benign" $\rightarrow$ 0, "DDoS" $\rightarrow$ 1). Each unique category was assigned an integer, making the labels compatible with machine learning algorithms. This transformation produced a label vector with numerical values, enabling efficient processing.

2.1.6 Feature Standardization

To standardize the feature space, each feature was normalized to have a mean of 0 and a standard deviation of 1 using z-score normalization:

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

where x is the original feature value, μ is the mean, and σ is the standard deviation. Standardization ensures that features with different scales or units do not disproportionately influence the learning process, which is particularly critical for algorithms sensitive to feature magnitude, such as neural networks.

2.2 Characteristics of the Processed Dataset

After processing, the dataset exhibited the following properties:

- **Feature Space:** 78 standardized numerical features, each scaled to a mean of 0 and a standard deviation of 1.

- **Label Distribution:** The dataset contains a mix of benign and attack flows, with multiple attack types represented. This diversity enables robust evaluation of the model's performance across different attack scenarios.
- **Subset Sizes:**
  - Training Set: **1,696,725 samples**
  - Validation Set: **565,575 samples**
  - Test Set: **565,576 samples**

  The splitting was performed using stratified sampling to preserve the distribution of classes across subsets, ensuring that all attack types are represented proportionally.

3 Model

This section describes the proposed neural network-based incremental learning model, its architecture, and the strategies to address adversarial attacks and catastrophic forgetting.

3.1 Model Overview

The model is designed for detecting Distributed Denial-of-Service (DDoS) attacks using the CIC-DDoS2019 dataset. It incorporates two key components:

1. **Feedforward Neural Network:** A fully connected neural network (consisting of layers `fc1`, `fc2`, and `fc3`) to classify network traffic.
2. **Adversarial Mitigation Strategies:** Mechanisms to counteract the effects of adversarial data poisoning and catastrophic forgetting, including a replay mechanism.

3.2 Neural Network Architecture

The neural network structure includes three layers:

- **Input Layer (`input`):** Accepts the standardized 78-dimensional feature vector.
- **Hidden Layers:**
  - `fc1`**:** A fully connected layer with 256 neurons and ReLU activation.
  - `fc2`**:** A fully connected layer with 128 neurons and ReLU activation.
- **Output Layer (`fc3`):** A fully connected layer producing logits for C classes, where C represents the number of attack types and benign traffic. During evaluation, logits are converted to probabilities using the softmax function.

3.2.1 Mathematical Description

The neural network performs the following computations:

$$\mathbf{h}_1 = \text{ReLU}(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1), \quad \text{(Layer: } \texttt{fc1}\text{)} \tag{2}$$

$$\mathbf{h}_2 = \text{ReLU}(\mathbf{h}_1\mathbf{W}_2 + \mathbf{b}_2), \quad \text{(Layer: } \texttt{fc2}\text{)} \tag{3}$$

$$\mathbf{y} = \mathbf{h}_2\mathbf{W}_3 + \mathbf{b}_3, \quad \text{(Layer: } \texttt{fc3}\text{)} \tag{4}$$

where $\mathbf{x}$ is the input, $\mathbf{W}_i$ and $\mathbf{b}_i$ are the weights and biases of layer i, $\mathbf{h}_i$ represents the output of hidden layer i, and $\mathbf{y}$ is the final output.

3.3 Adversarial Data Poisoning and Replay Mechanism

The robustness of the proposed model was evaluated under challenging scenarios involving adversarial attacks and catastrophic forgetting. These scenarios were addressed through two critical components: **Adversarial Data Poisoning** and the **Replay Mechanism**.

3.3.1 Adversarial Data Poisoning

Adversarial data poisoning introduces corrupted or adversarial samples into the training data to simulate real-world attack scenarios. The goal of these attacks is to degrade the model's ability to detect threats or to bias the model against specific classes. Three types of poisoning strategies were implemented:

- **Feature Perturbation:** [2] Introduces random noise to input features, causing the model to learn spurious patterns. This simulates scenarios where attackers manipulate traffic patterns to evade detection systems.
- **Label Flipping:** [1] Modifies the ground-truth labels for a subset of the training data. For instance, benign samples may be mislabeled as attacks, or vice versa. This strategy aims to confuse the model and reduce classification accuracy.
- **PGD Attack (Projected Gradient Descent):** [3] Generates adversarial examples by iteratively maximizing the model's loss with respect to a perturbation constraint. Formally, the perturbed input $\mathbf{x}'$ is computed as:

  $$\mathbf{x}' = \text{clip}_{\mathbf{x},\varepsilon}\big(\mathbf{x} + \alpha \cdot \text{sign}(\nabla_{\mathbf{x}}\mathscr{L})\big)$$

  where ε is the maximum allowable perturbation, α is the step size, and $\mathscr{L}$ is the loss function. This approach simulates sophisticated adversarial attacks aimed at bypassing the detection system.

Poisoning Rate: The fraction of the training data that is adversarially modified is controlled by the poisoning rate $r \in [0,1]$. Experiments were conducted for varying poisoning rates to evaluate their impact on model performance.

3.3.2 Impact of Adversarial Data Poisoning

Adversarial data poisoning directly challenges the model's robustness by degrading its ability to generalize and accurately classify traffic patterns. By progressively increasing the poisoning rate, the model's performance metrics such as accuracy, precision, recall, and F1-score were observed to decline. This analysis helps to identify vulnerabilities and limitations in the current system.

3.3.3 Replay Mechanism

To address the issue of **catastrophic forgetting**, a replay mechanism was implemented. Catastrophic forgetting occurs when the model overwrites previously learned knowledge while adapting to new data streams. The replay mechanism mitigates this effect by reintroducing a subset of previously seen data during training.

Key Components of Replay Mechanism:

- **Replay Buffer:** Stores a fixed fraction of the previously seen data. This buffer is updated dynamically, ensuring that representative samples from earlier data streams are retained.
- **Replay Training:** During each training epoch, the replay buffer is combined with the current training batch. The combined data is used to update the model, ensuring that both old and new patterns are retained.

Training Process with Replay: Let $\mathbf{X}_{\text{new}}$ and $\mathbf{Y}_{\text{new}}$ represent the current training data and labels, while $\mathbf{X}_{\text{replay}}$ and $\mathbf{Y}_{\text{replay}}$ denote the data and labels from the replay buffer. The replay training process involves:

1. Sampling a batch of size N from $\mathbf{X}_{\text{new}}$ and $\mathbf{Y}_{\text{new}}$.
2. Sampling a batch of size M from $\mathbf{X}_{\text{replay}}$ and $\mathbf{Y}_{\text{replay}}$.
3. Combining the two batches to form the training set for that iteration:

$$\mathbf{X}_{\text{train}} = \mathbf{X}_{\text{new}} \cup \mathbf{X}_{\text{replay}}, \quad \mathbf{Y}_{\text{train}} = \mathbf{Y}_{\text{new}} \cup \mathbf{Y}_{\text{replay}}$$

This ensures that the model retains its ability to detect previously learned attack patterns while adapting to new threats.

3.3.4 Evaluation of Replay Mechanism

The effectiveness of the replay mechanism was evaluated by comparing model performance before and after replay training. Metrics such as accuracy, F1-score, and confusion matrices were analyzed. Results showed a significant reduction in catastrophic forgetting, as the replay mechanism preserved the model's ability to recall older patterns while learning new ones.

3.4 Adversarial Poisoning and Replay Workflow

Figure 1 illustrates the integration of adversarial poisoning and replay mechanisms into the training workflow.

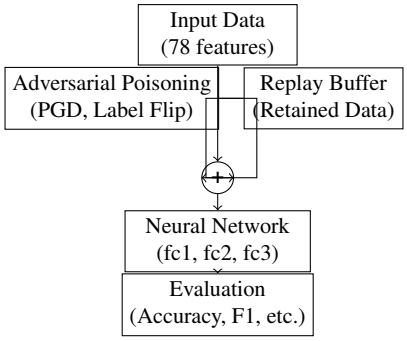


Figure 1: Workflow of Adversarial Poisoning and Replay Mechanism.

3.5 Model Workflow

The complete workflow of the model is depicted in Figure 2. The network structure explicitly includes layers `fc1`, `fc2`, and `fc3`, along with adversarial poisoning and replay mechanisms.

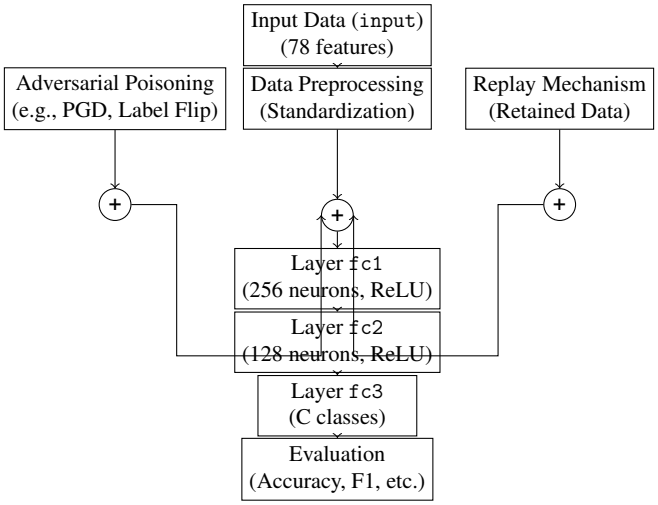


Figure 2: Workflow of the Incremental Learning Model with Explicit Layers (`fc1`, `fc2`, `fc3`).

3.6 Evaluation Metrics

The model's performance is evaluated using the following metrics:

- **Accuracy:** Measures the proportion of correctly classified samples.
- **Precision:** Evaluates the proportion of true positives among all predicted positives.
- **Recall:** Measures the proportion of true positives among actual positives.
- **F1-Score:** Harmonic mean of precision and recall
- **Confusion Matrix:** Provides a detailed breakdown of true positives, false positives, true negatives, and false negatives.

4 Analysis Results

This section provides a detailed analysis of the experimental results, focusing on the impact of various poisoning strategies on model performance and the effectiveness of the replay mechanism.

4.1 Overview of Results

The experiments assessed the model's robustness under six distinct poisoning strategies: *Feature Perturbation*, *Label Flip*, *Logic Disruption*, *Malicious Pattern*, *PGD Attack*, and *Combined*. Each strategy presented unique challenges, targeting different aspects of the model's learning process. The replay mechanism was evaluated as a mitigation strategy, demonstrating varying levels of effectiveness across poisoning rates.

Table 1 summarizes the overall impact of each poisoning strategy and the recovery performance of the replay mechanism.

Table 1: Impact and Recovery Summary for Poisoning Strategies

Strategy	**Attack Strength**	**Replay Effectiveness**	**Key Observations**
Feature Perturbation	Moderate	Moderate	Replay mitigates performance loss at low poisoning rates but less effective at higher rates.
Label Flip	High	Limited	Severe degradation at high poisoning rates; replay shows minimal recovery in extreme cases.
Logic Disruption	Low	Negligible	Minimal performance impact; the model shows strong robustness to this strategy.
Malicious Pattern	High	Low-Moderate	Effective recovery at low rates; limited impact at high poisoning rates.
PGD Attack	Moderate-High	Limited	Replay shows minor recovery under low poisoning rates but is ineffective at high rates.
Combined	Very High	Moderate at low rates	Highly destructive; replay provides significant recovery at moderate rates but fails at high rates.

4.1.1 Comprehensive Summary of Results

Table 2 consolidates the performance metrics for all poisoning strategies across varying poisoning rates. It provides a clear overview of the impact on accuracy and F1 scores before and after applying the replay mechanism.

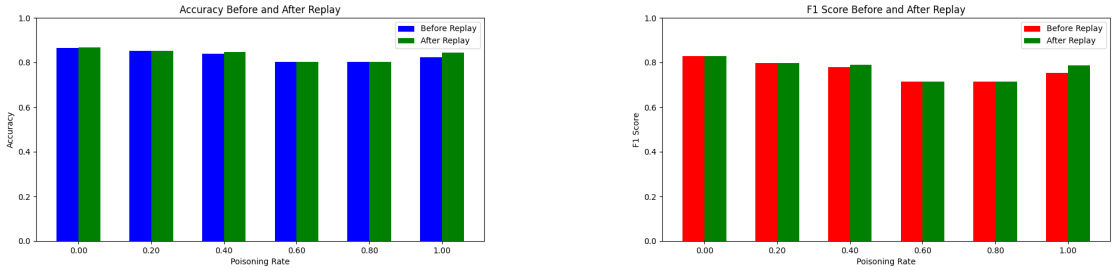
Table 2: Summary of Results for All Poisoning Strategies

Strategy	Poisoning Rate	Accuracy Before Replay	Accuracy After Replay	F1 Score Before Replay	F1 Score After Replay
Feature Perturbation	0.0	0.866	0.867	0.828	0.829
Feature Perturbation	0.4	0.839	0.847	0.780	0.790
Label Flip	0.6	0.008	0.160	0.014	0.256
Logic Disruption	0.8	0.852	0.852	0.796	0.796
Malicious Pattern	0.4	0.849	0.851	0.794	0.795
PGD Attack	0.8	0.832	0.835	0.770	0.775
Combined	0.4	0.297	0.600	0.442	0.684

4.2 Detailed Strategy Analysis

4.2.1 Feature Perturbation

Feature Perturbation introduces random noise into the input features. As shown in Figure 3, the replay mechanism successfully mitigates performance degradation at low poisoning rates (Poisoning Rate ≤ 0.4), with improvements in accuracy and F1 scores.

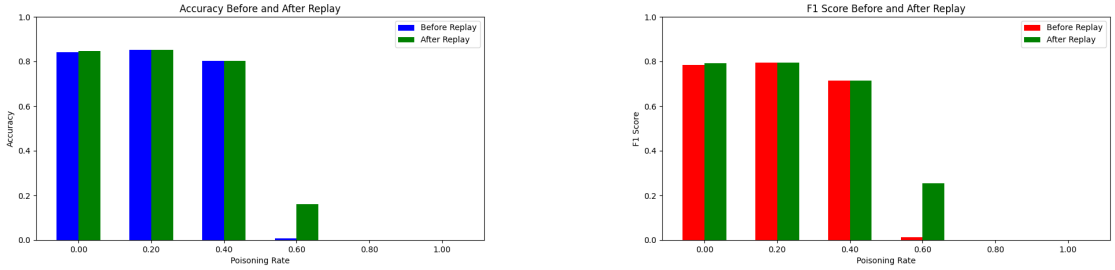


(a) Accuracy Comparison. (b) F1 Score Comparison.

Figure 3: Feature Perturbation: Performance before and after replay.

4.2.2 Label Flip

Label Flip significantly impacts model performance by altering the ground-truth labels. Figure 4 illustrates a drastic decline in accuracy and F1 scores at high poisoning rates (Poisoning Rate ≥ 0.6). Replay showed limited recovery, particularly at extreme poisoning levels.



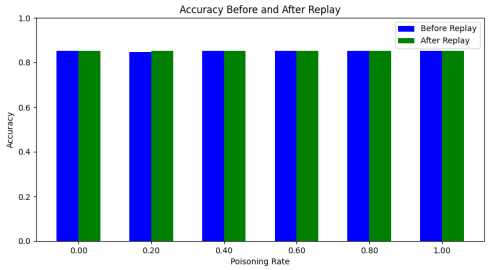
(a) Accuracy Comparison. (b) F1 Score Comparison.

Figure 4: Label Flip: Performance before and after replay.

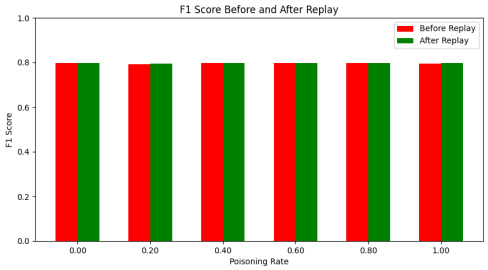
4.2.3 Logic Disruption[4]

Logic Disruption disrupts feature relationships but had minimal impact on model performance, as seen in Figure 5. Replay's effect was negligible due to the model's inherent

robustness against this attack.



(a) Accuracy Comparison.

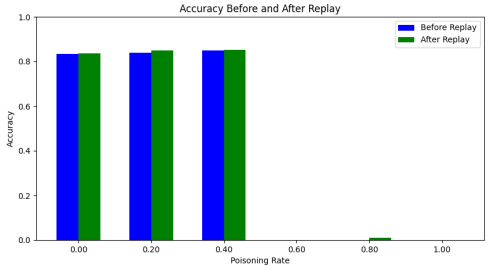


(b) F1 Score Comparison.

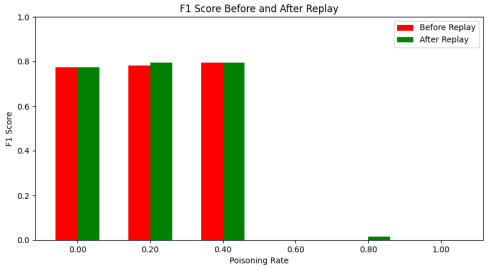
Figure 5: Logic Disruption: Performance before and after replay.

4.2.4 Malicious Pattern[5]

Malicious Pattern injected artificial patterns into the data, causing significant performance degradation at higher poisoning rates. As shown in Figure 6, replay mitigated performance loss at low poisoning rates but failed to restore performance at high rates.



(a) Accuracy Comparison.

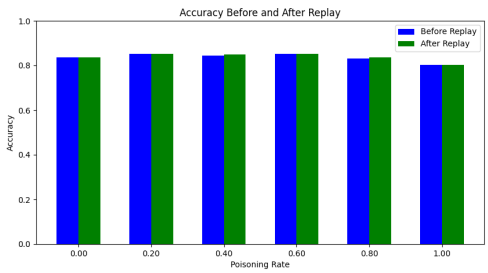


(b) F1 Score Comparison.

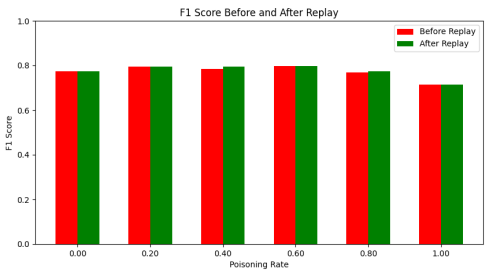
Figure 6: Malicious Pattern: Performance before and after replay.

4.2.5 PGD Attack

The PGD Attack generated adversarial samples to maximize loss, significantly degrading performance at high poisoning rates. Replay showed limited recovery, as seen in Figure 7.



(a) Accuracy Comparison.

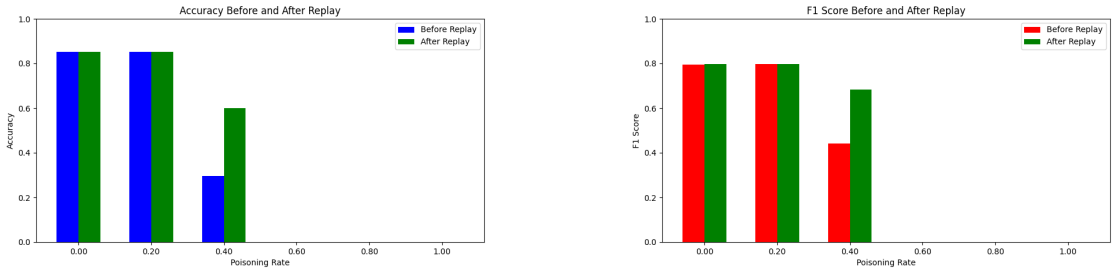


(b) F1 Score Comparison.

Figure 7: PGD Attack: Performance before and after replay.

4.2.6 Combined Strategy

The Combined strategy demonstrated the most destructive impact by incorporating multiple attack methods. Figure 8 highlights significant performance degradation at higher poisoning rates (Poisoning Rate ≥ 0.6). Replay provided moderate recovery at Poisoning Rate $= 0.4$, where accuracy improved from 29.68% to 60.00% and F1 score from 44.22% to 68.39%.



(a) Accuracy Comparison.

(b) F1 Score Comparison.

Figure 8: Combined Strategy: Performance before and after replay.

4.3 Replay Mechanism: Effectiveness and Limitations

The replay mechanism exhibited varying levels of success based on the nature of the poisoning strategy and the poisoning rate:

- **Low Poisoning Rates (Poisoning Rate ≤ 0.4)**: Replay effectively mitigated performance degradation across most strategies. Accuracy and F1 score showed consistent improvements post-replay.
- **High Poisoning Rates (Poisoning Rate ≥ 0.6)**: Replay's effectiveness declined significantly. For destructive strategies such as *Label Flip* and *Combined*, recovery was negligible.
- **Strategy-Dependent Behavior**: Replay was particularly effective for *Feature Perturbation*, moderately effective for *Malicious Pattern*, and minimally effective for *PGD Attack* and *Label Flip*.

4.4 Consolidated Observations

From the consolidated analysis:

- **Replay Mechanism Effectiveness:** Replay demonstrates high effectiveness for low to moderate poisoning rates (Poisoning Rate ≤ 0.4), particularly for strategies such as *Feature Perturbation* and *Malicious Pattern*.
- **High Poisoning Rates (Poisoning Rate ≥ 0.6):** Replay's impact diminishes for destructive strategies such as *Label Flip* and *Combined*, with accuracy and F1 scores approaching zero.
- **Strategy-Specific Insights:** *Logic Disruption* exhibited minimal performance degradation, demonstrating model robustness. *Combined* was the most destructive, with partial recovery at moderate poisoning rates.

- **Practical Implications**: Replay mechanisms alone are insufficient to counteract severe adversarial attacks, highlighting the need for more robust defenses.

5 Conclusion

In this study, we examined the robustness of a class-incremental learning system against various data poisoning strategies and evaluated the effectiveness of a replay mechanism in mitigating their impact. The CIC-DDoS2019 dataset was prepared through extensive preprocessing, enabling reliable experimentation. Six distinct poisoning strategies revealed varying vulnerabilities in the model, with Replay demonstrating strong recovery at low to moderate poisoning rates but struggling against highly destructive attacks such as Label Flip and Combined strategies. Overall, the findings underscore the importance of robust defenses in incremental learning systems, highlighting Replay's potential as a mitigation strategy while emphasizing the need for further advancements to address severe adversarial scenarios effectively.

References

[1] Lakshmeeswari Gondi, Swathi Sambangi, P Kundana Priya, and S Sharika Anjum. A machine learning approach for ddos attack detection in cic-ddos2019 dataset using multiple linear regression algorithm. In *XVIII International Conference on Data Science and Intelligent Analysis of Information*, pages 393–403. Springer, 2023.

[2] Chenming Li, Daoan Zhang, Wenjian Huang, and Jianguo Zhang. Cross contrasting feature perturbation for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1327–1337, 2023.

[3] Hetvi Waghela, Jaydip Sen, and Sneha Rakshit. Robust image classification: Defensive strategies against fgsm and pgd adversarial attacks. *arXiv preprint arXiv:2408.13274*, 2024.

[4] Petri Vähäkainu, Martti Lehto, and Antti Kariluoto. Adversarial poisoning attack's impact on prediction functionality of ml-based feedback loop system in cyber-physical context. In *ICCWS 2021 16th International Conference on Cyber Warfare and Security*, page 373. Academic Conferences Limited, 2021.

[5] Jian Chen, Xuxin Zhang, Rui Zhang, Chen Wang, and Ling Liu. De-pois: An attack-agnostic defense against data poisoning attacks. *IEEE Transactions on Information Forensics and Security*, 16:3412–3425, 2021.