Evaluating Evasion in Network Intrusion Detection Systems

Final Project Report

Course: Introduction to Data Security
Semester: Spring 2025

Authors:
Miguel Pinto
Mátyás Szikra
Ahmed Lotfy

Eötvös Loránd University (ELTE)
Submission Date: May 4, 2025

Table of Contents

1.1 Project Motivation and Objectives

Evasive cyberattacks are increasingly designed to bypass traditional network intrusion detection systems (NIDS) by mimicking normal traffic or operating at low, undetectable rates. This project explores the effectiveness of machine learning-based NIDS models when confronted with such evasive behavior.

The main objectives are:

- To develop supervised and unsupervised detection models using benchmark data;
- To generate and use real and synthetic network traffic containing evasive attacks;
- To evaluate model performance in controlled adversarial scenarios;
- To identify strengths and limitations in detecting stealthy intrusions.

1.2 Challenges in Network Intrusion Detection, Particularly Evasion

Detecting evasive attacks is difficult because they are subtle, resemble normal activity, and often avoid triggering thresholds used by standard IDS models. Raising detection sensitivity can lead to high false-positive rates, while lowering it risks missing stealthy threats.

Key challenges include:

- Differentiating low-signal attacks from normal traffic;
- Generalizing to unseen or modified evasion techniques;
- Maintaining performance under incomplete or ambiguous data conditions.

1.3 Overview of Approach and Project Structure

This project follows a multi-stage approach to evaluate how well machine learning-based network intrusion detection systems (NIDS) perform against evasive attacks. The process begins with the use of a benchmark dataset to establish baseline models and progresses into real-world testing using generated network traffic, including deliberately stealthy attacks.

The structure of the project is as follows:

- **Model Development:** Initial experiments use the CICIDS2017 dataset to train both unsupervised (Isolation Forest) and supervised (Random Forest) models. After early testing, the project pivots to a supervised approach based on limitations observed in anomaly detection performance.

- **Traffic Generation:** Real and synthetic network traffic is generated in a controlled virtual environment. The attack traffic focuses on low-rate nmap port scans designed to evade detection. Benign traffic is also collected to simulate realistic network conditions.
- **Adversarial Evaluation:** The trained models are tested against whitebox, greybox, and blackbox evasion scenarios. Performance is measured to identify vulnerabilities, and models are iteratively updated based on the most effective attack samples.
- **Analysis and Reflection:** The final stage analyzes the detection results, highlighting the effectiveness and limitations of the chosen methods, and discusses opportunities for improvement in future work.

This structure ensures a comprehensive evaluation of detection strategies against evasive techniques, with both theoretical and practical components informing the results.

1.4 Reference Context Across Phases

Key research was considered throughout the project to guide design and methodology:

- **Model Selection:** Isolation Forest and Random Forest were chosen with awareness of foundational works by Liu et al. (2012) and Breiman (2001), along with applied uses in intrusion detection (e.g., Chua et al., 2024; Wu et al., 2022).
- **Traffic and Attack Generation:** Evasion techniques were aligned with known scan behavior described by Ring et al. (2018) and Dabbagh et al. (2011), focusing on slow, stealthy probes.
- **Adversarial Testing:** The structure of whitebox, greybox, and blackbox evaluations was informed by Yazdanpour et al. (2023) and Huang et al. (2019), which outline threat models based on attacker knowledge.
- **Robustness and Evasion Defense:** Final tuning strategies drew on ideas from Peng et al. (2020) and Alshahrani et al. (2022), who explored retraining and synthetic evasion traffic.

These references supported decisions in each phase and ensured alignment with current research in adversarial NIDS.

2. Dataset-Based Modeling

2.1 Exploration of CICIDS2017 Data

The CICIDS2017 dataset serves as the foundation for the initial modeling phase. It contains labeled network flow records simulating real-world traffic, including both benign activity and a variety of attack types. For this project, several daily CSV logs were combined to form a unified dataset.

Exploration focused on understanding feature distributions, class balance, and data quality. Non-informative fields (such as IP addresses and timestamps) were excluded, and labels were simplified into binary classes: *benign* and *attack*. This setup provided a consistent structure for both unsupervised and supervised experiments in the modeling pipeline.

2.2 Unsupervised Modeling with Isolation Forest

Isolation Forest was used as the first modeling approach to detect anomalies without relying on labeled attack data. The model was trained only on benign flows to establish a baseline for normal traffic behavior. Predictions were made on the entire dataset, with raw anomaly scores converted into binary decisions using a tunable threshold.

Threshold selection aimed to **maximize recall**—ensuring as many attacks as possible were detected—**without sacrificing overall balance**. This was done by analyzing F1-scores across a range of thresholds to identify the point with the best trade-off between true positive rate and false positives.

While the model successfully identified a significant portion of attack traffic, it also misclassified a notable number of benign samples. These false positives highlighted the limits of using anomaly detection in a setting where some legitimate behaviors may appear atypical.

Classification Report:

	precision	recall	f1-score	support
Benign	0.87	0.92	0.89	2271320
Attack	0.57	0.45	0.50	556556
accuracy			0.82	2827876
macro avg	0.72	0.68	0.70	2827876
weighted avg	0.81	0.82	0.82	2827876

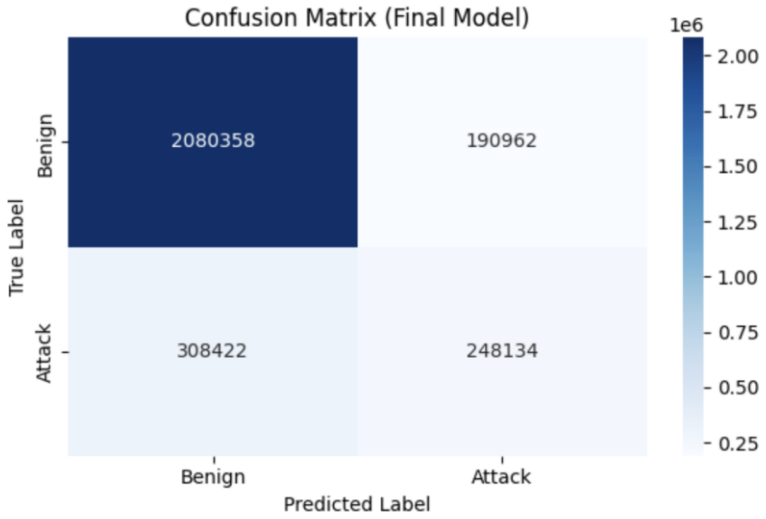


Figure 1. *Classification Report after the first training*

```
=== Classification Report ===
              precision    recall  f1-score   support

      Benign       0.94      0.71      0.81   2271320
      Attack       0.41      0.82      0.55    556556

    accuracy                           0.73   2827876
   macro avg       0.68      0.77      0.68   2827876
weighted avg       0.84      0.73      0.76   2827876


=== Detailed Statistics ===
True Negatives (Benign correctly identified): 1,603,676
False Positives (Benign predicted as attack): 667,644
False Negatives (Missed attacks): 97,582
True Positives (Attacks detected): 458,974

Total Samples: 2,827,876
Attack Recall: 0.8247
Benign Recall: 0.7061
False Positive Rate (Benign flagged): 0.2939
Overall Precision: 0.4074
Overall F1-Score: 0.5454
Overall Accuracy: 0.7294
```

Figure 2. *Classification Report after the aggressive recall tuning*

2.3 Evaluation on Real Traffic and Limitations of Unsupervised Detection

Following initial success on synthetic and benchmark data, the aggressively tuned Isolation Forest model was evaluated on real-world benign and attack traffic generated in a controlled environment. The aim was to test the model's practical utility in a more realistic setting.

The model, previously trained only on benign CICIDS2017 traffic, was applied to real benign flows. Despite careful threshold tuning to maximize recall while maintaining balance, the model misclassified the majority of benign traffic as attacks. This excessive false positive rate rendered it unusable in a real deployment context.

Key Observations:

- **On benign-only traffic**, the model predicted nearly all samples as attacks, regardless of their actual content. This indicated that the model had **overfit to the training distribution** and was unable to generalize to different benign patterns.
- **Score distributions showed minimal separation** between benign and malicious classes under real conditions, reducing the threshold's discriminative power.
- **Even after incorporating labeled real traffic for evaluation**, the model produced **imbalanced predictions**, heavily skewed toward the "attack" label.

This failure revealed a critical limitation of the unsupervised approach: its reliance on a narrow definition of "normal" behavior makes it overly sensitive when faced with real,

diverse traffic—much of which is legitimate but not identical to the training data. These findings led to the decision to abandon Isolation Forest in favor of a supervised learning model, which could leverage labeled examples to better distinguish between benign and malicious behavior.

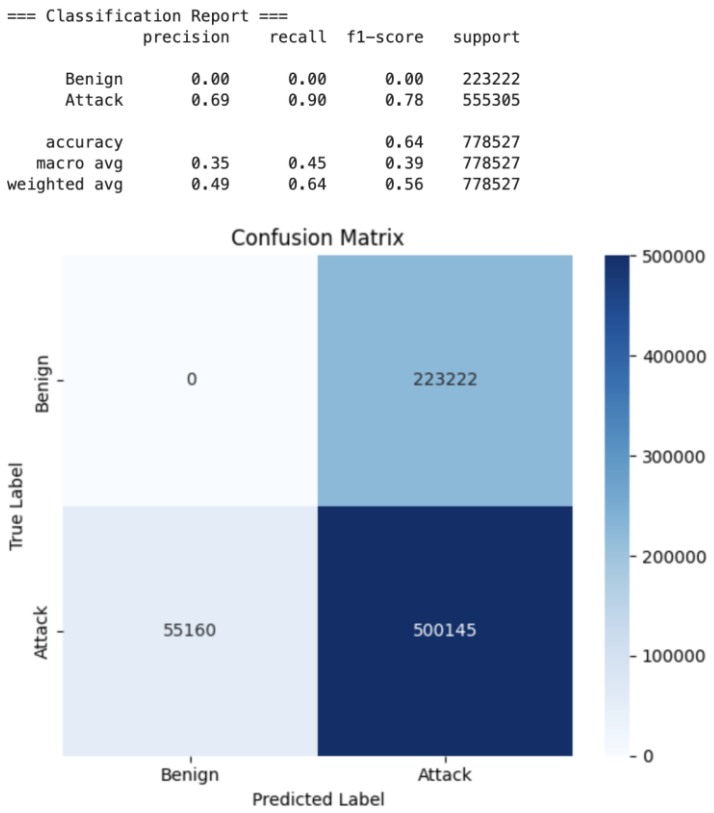


Figure 3. *Classification report on real data sample*

2.4 Transition to Supervised Learning with Random Forest

After identifying the limitations of unsupervised anomaly detection, the project adopted a supervised learning approach using a Random Forest classifier. This method was chosen for its robustness, ability to handle high-dimensional data, and natural support for multi-feature, labeled input.

The classifier was trained on the preprocessed CICIDS2017 dataset, which includes binary labels indicating whether each network flow is benign or an attack. A standard 80/20 train-test split was used with stratified sampling to preserve class distribution. The model was trained using 100 decision trees and evaluated on the test set using precision, recall, F1-score, and a confusion matrix.

The results demonstrated significant improvements in both accuracy and balance between detecting attacks and minimizing false positives—addressing the core issues encountered in the unsupervised phase.

Key Advantages Observed:

- **High precision and recall** for both benign and attack classes
- **Stable performance** across diverse traffic patterns, thanks to labeled supervision
- **Significant reduction in false positives**, improving practical usability

```
Evaluating on test set...
              precision    recall  f1-score   support

      Benign       1.00      1.00      1.00    454265
      Attack       1.00      1.00      1.00    111311

    accuracy                           1.00    565576
   macro avg       1.00      1.00      1.00    565576
weighted avg       1.00      1.00      1.00    565576
```

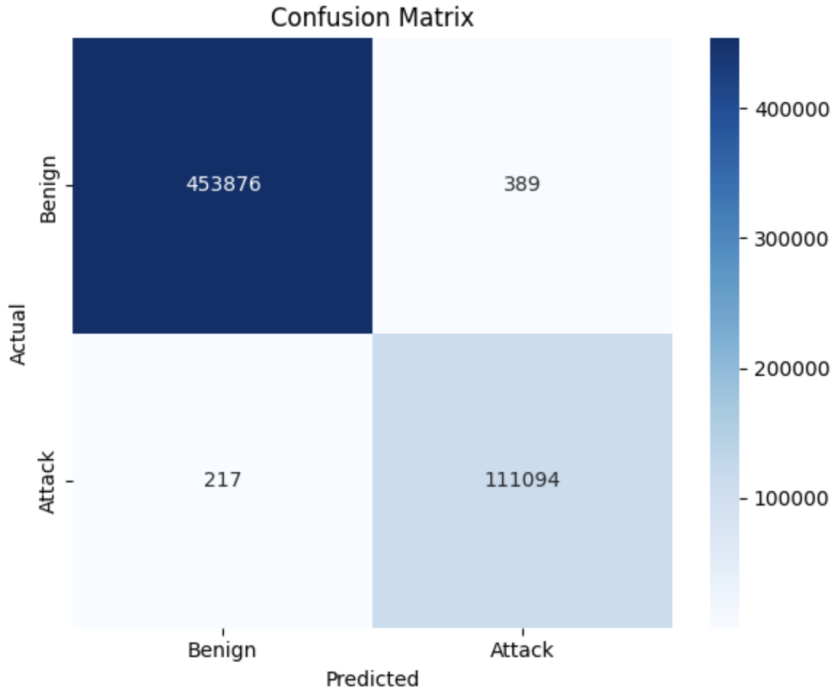


Figure 4. *Classification report after training the supervised model*

This supervised model now serves as a baseline for subsequent evaluations on real and adversarial traffic in the next stages of the project.

3. Real Traffic Generation and Feature Extraction

3.1 Virtual Testbed and Tooling

Traffic generation was conducted using a simple two-VM setup. A **Ubuntu VM** acted as the victim, generating benign traffic and hosting services. A separate **Kali Linux VM** was used to launch attacks, primarily using `nmap` for stealthy scanning.

Both machines were connected on a shared virtual network, allowing full packet capture at the victim's side. This setup provided a controlled environment for simulating realistic network behavior while keeping traffic isolated and reproducible.

Captured PCAPs were later used for feature extraction and labeling in the evaluation process.

3.2 Evasive Attack Design Using Nmap

To test the intrusion detection model against stealthy threats, a variety of **evasive port scans** were generated using nmap in the attacker VM. The objective was to simulate realistic adversarial behavior aimed at **evading detection** by reducing signature visibility and minimizing statistical anomalies.

Several parameters were varied to achieve evasiveness:

- **Timing and rate control**: Packet delays, low-frequency scans, and sparse probe distribution.
- **TCP flag manipulation**: Use of non-standard flag combinations to bypass signature-based rules.
- **Packet fragmentation and sequence obfuscation**: Techniques to confuse reassembly and detection.
- **TCP-level evasion tricks**: Adjustments in window size, sequence numbers, and randomization to mimic benign patterns.

These scans reflect a stealth-oriented attacker profile and were carefully captured for use in evaluation. In addition to nmap, **custom Python scripts were also used** to generate traffic with specific timing and structure constraints to further test detection robustness.

3.3 Benign Traffic Collection

To complement the attack data, a variety of **benign traffic samples** were collected under controlled conditions. These captures aimed to reflect normal system usage, network background activity, and different levels of load, helping the model distinguish real threats from legitimate behavior.

Benign traffic generation included:

- **Simulated user activity**, such as browsing, file access, and typical service interactions.
- **High-traffic scenarios** involving large data transfers or multiple concurrent connections.
- **Diverse protocol usage**, with traffic patterns intended to mirror a mix of common applications and services.

Several scripted sessions were executed to produce reproducible patterns across different datasets. All benign traffic was labeled accordingly and used in both model training and evaluation to ensure the system could maintain **low false positive rates** when exposed to varied legitimate behavior.

3.4 Feature Extraction and Dataset Preparation

To convert captured network traffic into a format suitable for machine learning, **custom Python scripts** were used to extract flow-based features from the raw PCAP files. These scripts were developed separately for attack and benign traffic, with each labeling the output accordingly for downstream use.

The extraction process included:

- **Flow segmentation** based on time, packet count, and idle intervals;
- **Feature calculation** such as packet size statistics, flow durations, inter-arrival times, burstiness, and directionality;
- **Label assignment**: attack traffic was labeled as 1, benign as 0.

The scripts produced structured CSV files consistent with CICIDS-style features, allowing seamless integration with the earlier preprocessing and model training pipeline. This approach ensured that the model could be evaluated on **real, diverse traffic** while maintaining consistency with its original training data structure.

4. Adversarial Testing and Model Evaluation

4.1 Evaluation on Real Traffic Samples

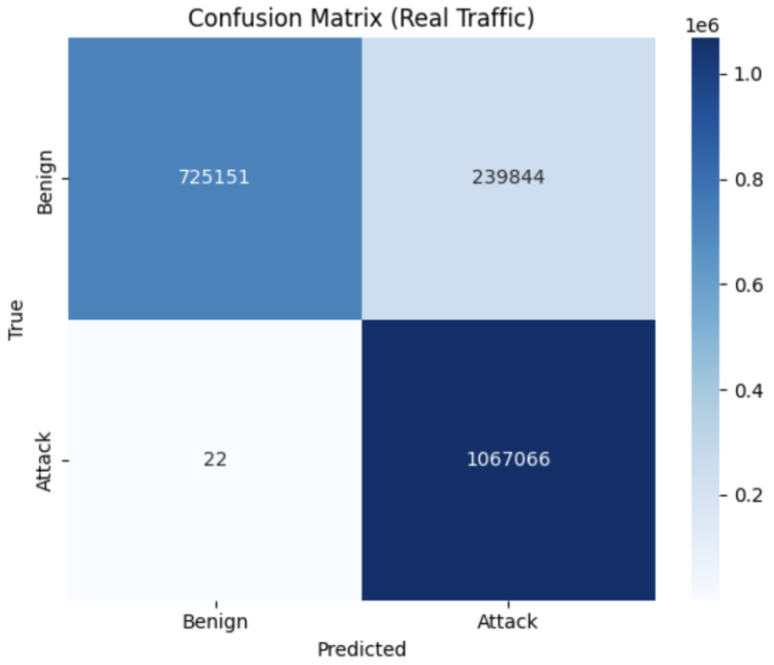
After training and tuning the supervised Random Forest model on labeled CICIDS-style data, the next step was to evaluate its effectiveness on **real benign and attack traffic** generated in the virtual testbed. This was essential to assess the model's generalization beyond synthetic or benchmark datasets.

Labeled real-world traffic was loaded, checked for feature consistency, and passed through the trained model. Predictions were compared to ground truth labels to calculate precision, recall, F1-score, and ROC AUC.

Key findings:

- The model maintained **high recall** on real attack traffic, correctly identifying most stealthy probes.
- **False positive rates were significantly lower** than in the unsupervised phase, indicating improved reliability on benign traffic.
- **Threshold tuning** (using predicted probabilities) helped optimize the balance between over-detection and under-detection.

These results confirmed that the model was able to **generalize effectively** to realistic, previously unseen traffic, a critical requirement for real-world NIDS deployment.



Classification Report on Real Traffic:

	precision	recall	f1-score	support
Benign	1.00	0.75	0.86	964995
Attack	0.82	1.00	0.90	1067088
accuracy			0.88	2032083
macro avg	0.91	0.88	0.88	2032083
weighted avg	0.90	0.88	0.88	2032083

Figure 5. *Classification report after training with real data samples*

4.2 Adversarial Attack Scenarios

To assess the robustness of the model under adversarial conditions, it was tested against a range of **evasion attacks** designed under varying levels of attacker knowledge. The goal was to simulate realistic threat models and determine how well the detection system performs when attackers actively attempt to bypass it.

The evaluation was structured into three threat models:

4.2.1 Whitebox Attacks

The following greybox attacks were tested:

White-box Attack 1: Simple Fixed Perturbation on Key Features
This basic attack uniformly reduces several known high-impact features—such as packet sizes, durations, and flow rates—by 5%. It simulates a naive adversary making broad, fixed changes without adaptation to context or feedback.

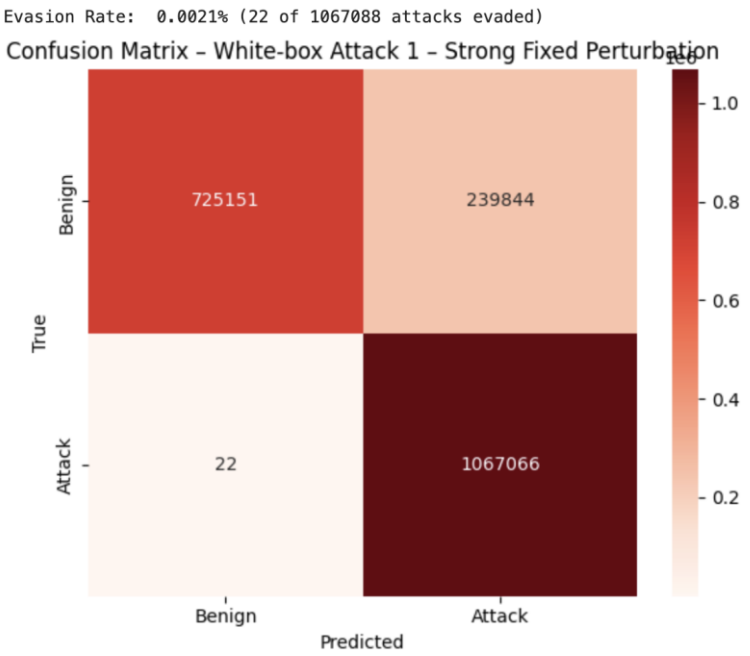


Figure 6. Detection Results – White-box Attack 1

White-box Attack 2: Adaptive Perturbation Based on Feature Magnitude
This attack scales perturbations depending on each feature's value: smaller features are altered more significantly, while larger ones are perturbed minimally. It reflects a more strategic adversary that adjusts changes based on the likelihood of detection.

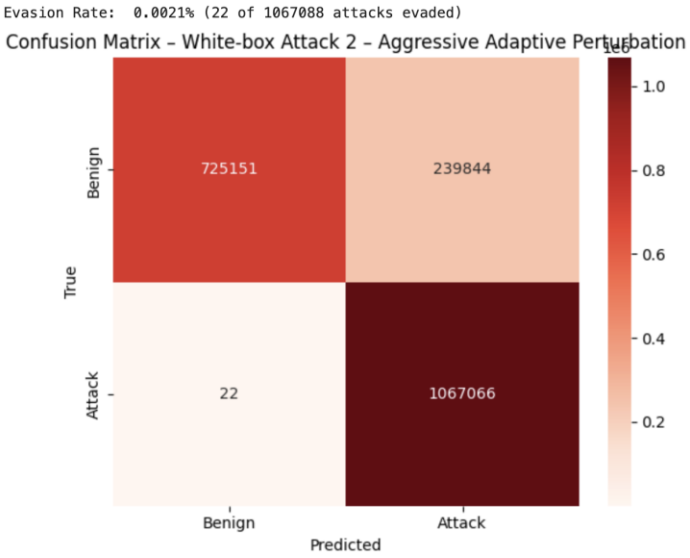


Figure 7. Detection Results – White-box Attack 2

White-box Attack 3: Feature Importance Guided Perturbation
The attacker uses model-derived feature importance rankings to focus changes on only the top features. These key fields are reduced by up to 40%, aiming to degrade detection while leaving the rest of the traffic unaltered.

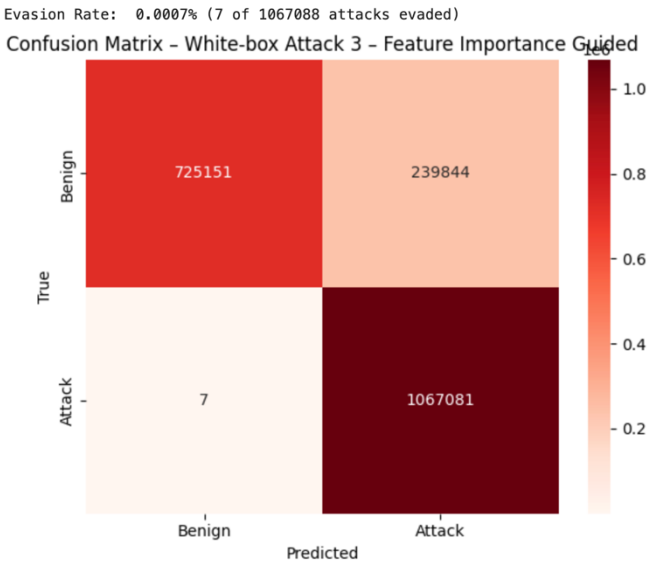


Figure 8. Detection Results – White-box Attack 3

White-box Attack 4: Partial Benign Mimicry (Top-K Features Only)
Only the top 5 most important features are adjusted to appear more like benign traffic. This lighter-touch attack blends realism with evasion, simulating a threat actor who wants to maintain functional payloads while avoiding detection.

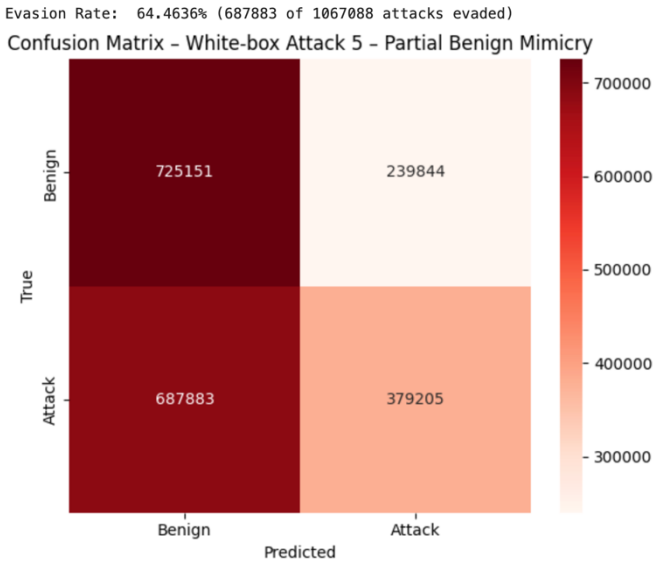


Figure 9. Detection Results – White-box Attack 4

White-box Attack 5: Benign Feature Distribution Mimicry
In the most comprehensive strategy, all numeric features are modified to mimic the statistical distribution (mean and standard deviation) of benign flows. Random noise is added to avoid overly uniform behavior, simulating a high-effort mimicry attack.

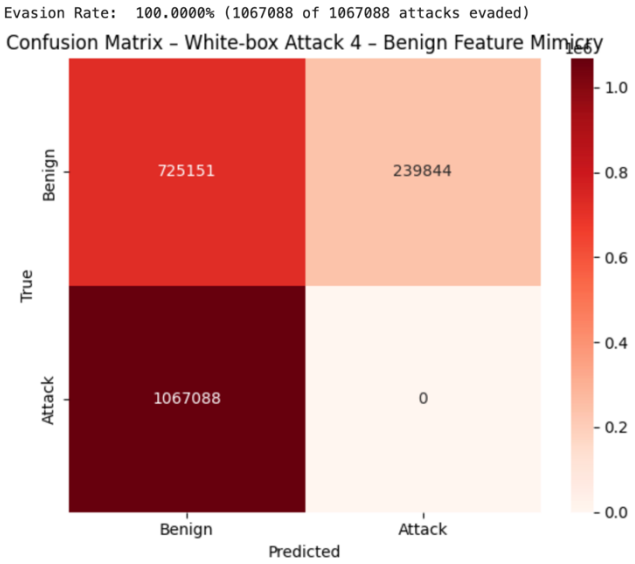


Figure 10. Detection Results – White-box Attack 5

These attacks illustrate how model-aware adversaries can systematically degrade IDS performance. As the level of mimicry and targeting increased, so did evasion success—highlighting the need for models that go beyond static, hand-engineered features.

4.2.2 Greybox Attacks

The following greybox attacks were tested:

Grey-box Attack 1: Query-Based Boundary Drift
Attack samples were gently shifted toward the mean values of the top 5 features from benign flows. The perturbation was minimal (2%), simulating an exploratory attacker attempting to slightly blend malicious traffic into the benign distribution without major distortion.

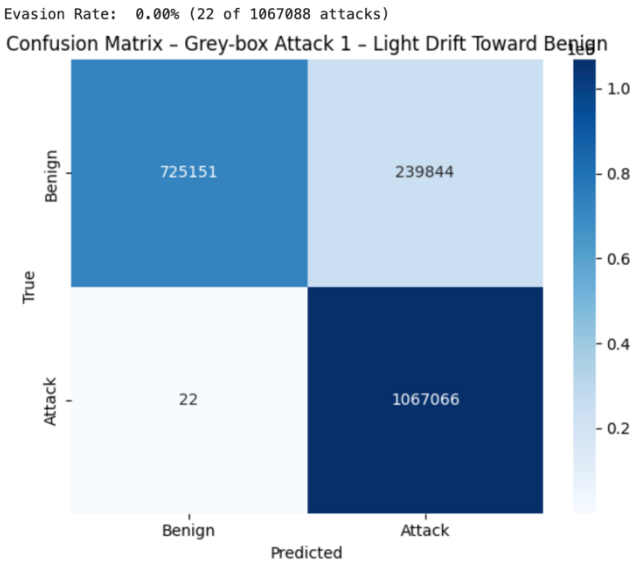


Figure 11. Detection Results – Grey-box Attack 1

Grey-box Attack 2: Benign Cluster Imitation (Top Features)
This attack replaced values in the top 3 most important features with those sampled directly from benign flows. A fourth feature was softly blended (90% original, 10% benign). This strategy reflects an attacker who has access to benign traffic and tries to replicate benign structure in high-value features.

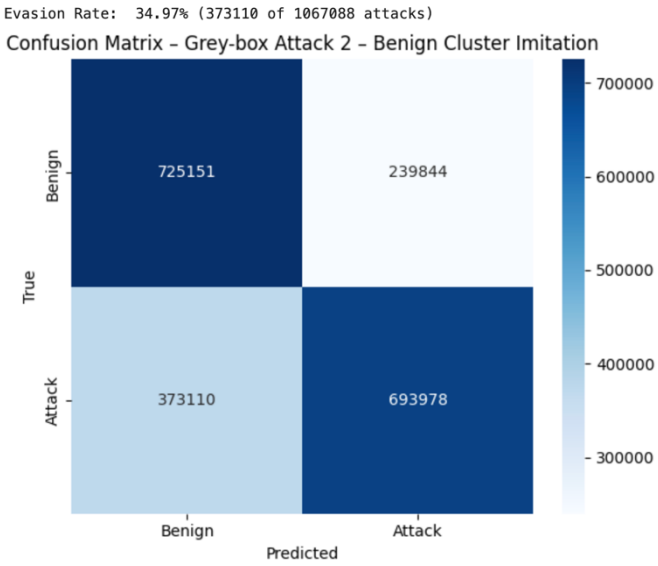


Figure 12. Detection Results – Grey-box Attack 2

Grey-box Attack 3: Surrogate Model Guided Evasion
The attacker trains a surrogate model using samples labeled as attack or benign, then identifies inputs that evade the surrogate. Selected feature values from these "surrogate-evading" samples are substituted into real attacks. This simulates transfer-based evasion, where the attacker crafts inputs on one model expecting them to succeed on another.

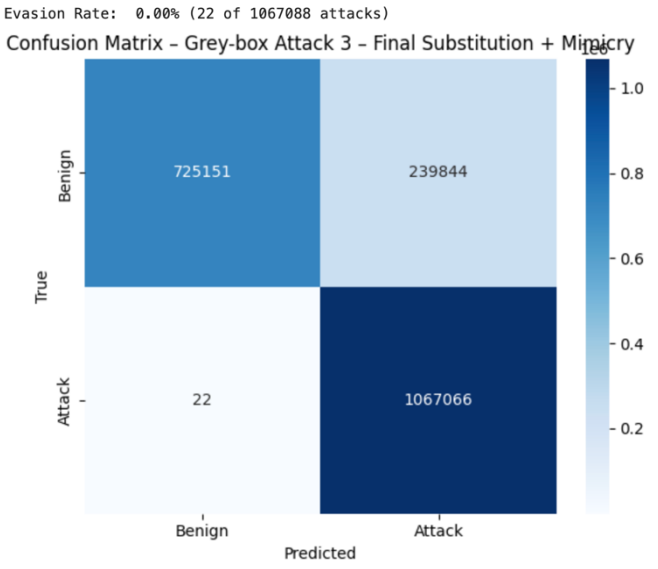


Figure 13. Detection Results – Grey-box Attack 3

Grey-box Attack 4: Class Blending (Benign + Attack)
In this attack, equal-sized pools of benign and attack flows are blended using a weighted average (70% benign, 30% attack). The resulting traffic is labeled as attack but appears statistically closer to benign patterns. This simulates a scenario where attackers embed malicious behavior within seemingly harmless traffic signatures.

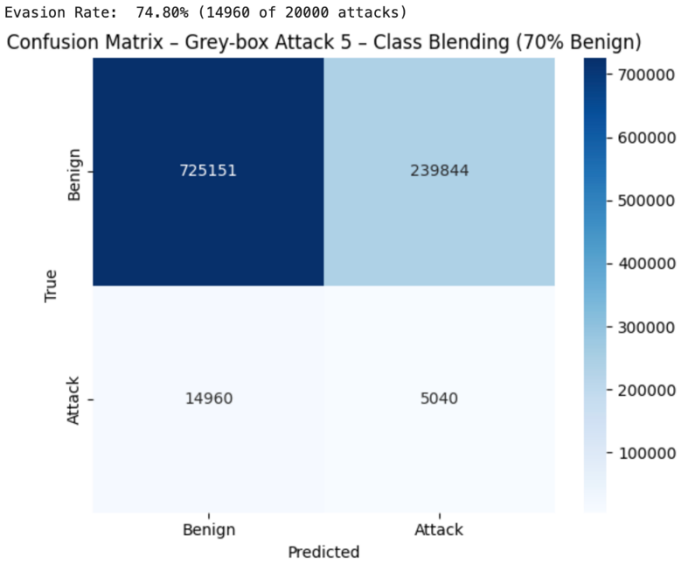


Figure 14. Detection Results – Grey-box Attack 4

These greybox attacks demonstrated varying degrees of success depending on how aggressively and precisely benign characteristics were mimicked. Even without full model access, attackers could achieve measurable evasion rates by leveraging query feedback and statistical profiling.

4.2.3 Blackbox Attacks

Three black-box evasion attacks were implemented:

Black-box Attack 1: Flag Suppression (Light)
This attack simulates an adversary who assumes that certain TCP flags (e.g., FIN, RST, SYN, ACK) are strong indicators of malicious traffic. Without model introspection, the attacker sets all such flags to zero in attack flows, attempting to make traffic appear neutral.

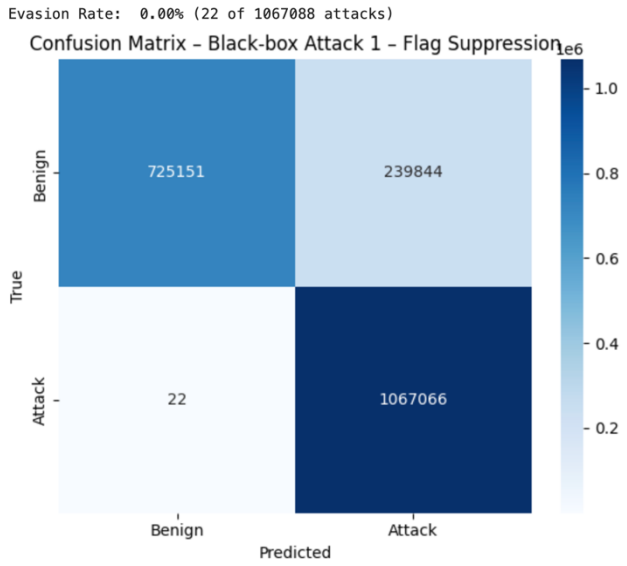


Figure 15. Detection Results – Black-box Attack 1

Black-box Attack 2: Full-Feature Blending (α = 0.10)
Each attack flow is blended with a benign flow at a low ratio (10% benign, 90% attack). The goal is to nudge traffic slightly toward benign characteristics while retaining most of the original behavior—creating a realistic evasion method with minimal distortion.

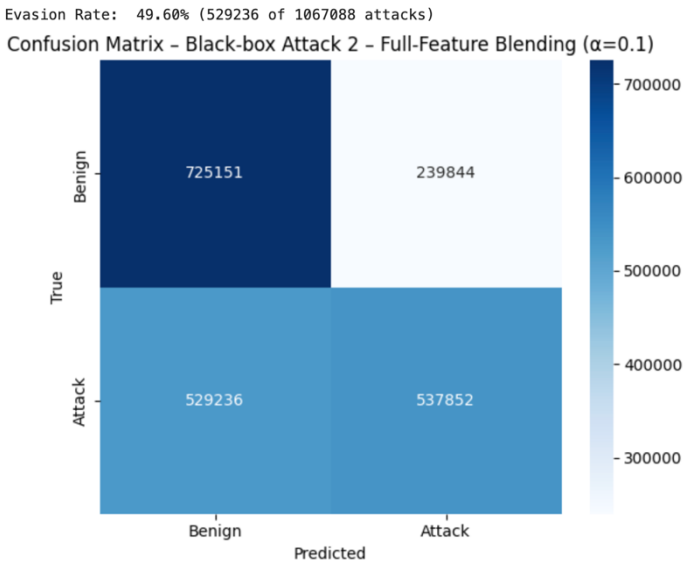


Figure 16. Detection Results – Black-box Attack 2

Black-box Attack 3: High-Intensity Class Blending (α = 0.80)
This more aggressive strategy blends attack flows with benign samples at a high ratio (80% benign, 20% attack). The resulting traffic appears highly benign and is harder to distinguish from legitimate activity. This represents a strong evasive attempt under blackbox constraints.

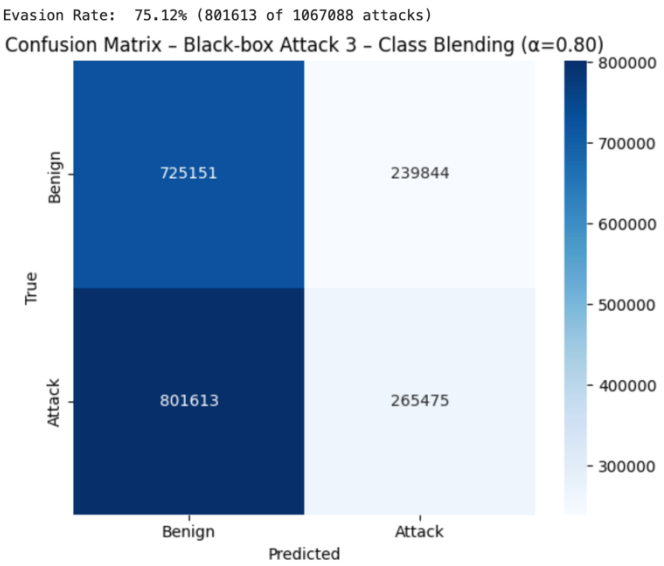


Figure 17. Detection Results – Black-box Attack 3

Although none of these attacks used model internals or surrogate models, they still achieved non-trivial evasion rates. Their simplicity also highlights the potential for **low-resource adversaries** to bypass detection when leveraging benign statistical patterns.

4.3 Final Evasive Attack and Model Retraining

After evaluating the model against multiple adversarial scenarios, a final training step was performed to improve detection of evasive behavior. A new Random Forest model was trained on a **combined dataset** consisting of:

- Clean real benign and attack flows,
- The strongest whitebox, greybox, and blackbox evasion samples generated during earlier testing.

The model architecture and hyperparameters were kept consistent with the baseline to focus solely on the impact of adversarial exposure.

```
Classification Report — Robust RF on Held-Out Real Traffic
              precision    recall  f1-score   support

      Benign       1.00      1.00      1.00    289499
      Attack       1.00      1.00      1.00    320126

    accuracy                           1.00    609625
   macro avg       1.00      1.00      1.00    609625
weighted avg       1.00      1.00      1.00    609625

Accuracy:     1.0000
Precision:    1.0000
Recall:       1.0000
F1-Score:     1.0000
ROC AUC:      1.0000
Evasion Rate: 0.00% (0 of 320126 attacks)
```

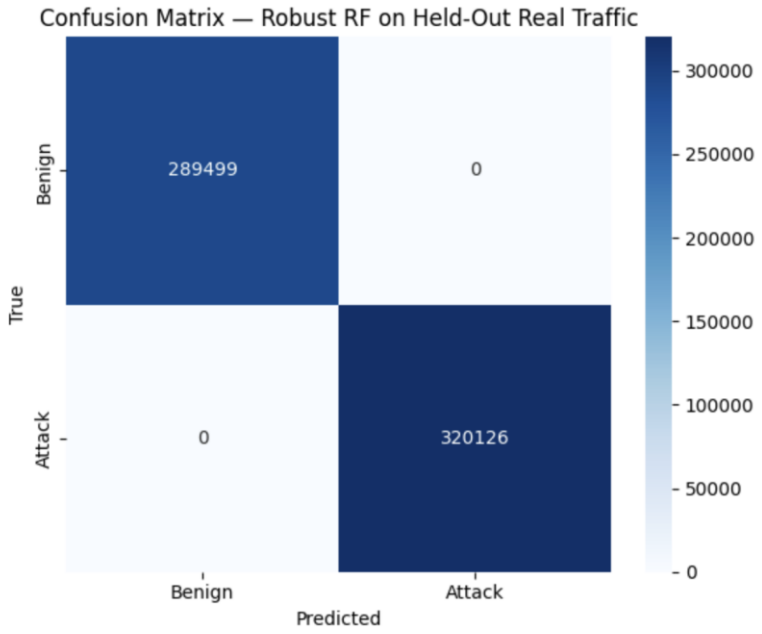


Figure 18. Confusion Matrix – Retrained Model on Clean Hold-Out Traffic

Evaluation on Adversarial Samples

The retrained model was tested on previously unseen adversarial examples. Performance improved across all three threat models, showing higher recall for evasive attacks without significantly increasing false positives.

Final Blackbox Evasion Strategy

A final, decision-based blackbox attack was executed to assess whether the retrained model could still be bypassed without internal knowledge. For each held-out attack flow:

1. The closest benign flow (Euclidean distance) was identified from the training set.
2. A **binary search** was performed to blend the attack and benign sample incrementally.
3. The goal was to find the **minimal blending ratio** that caused the attack to be misclassified as benign.

This method required only model outputs (0 or 1) and no feature importance or score feedback, making it realistic for a low-visibility adversary.

```
Classification Report — Nearest-Benign Blend Attack on N=1000
              precision    recall  f1-score   support

      Benign       1.00      1.00      1.00    289499
      Attack       0.00      0.00      0.00      1000

    accuracy                           1.00    290499
   macro avg       0.50      0.50      0.50    290499
weighted avg       0.99      1.00      0.99    290499

Accuracy:     0.9966
Precision:    0.0000
Recall:       0.0000
F1-Score:     0.0000
ROC AUC:      0.5460
Evasion Rate: 100.00% (1000 of 1000 attacks)
```

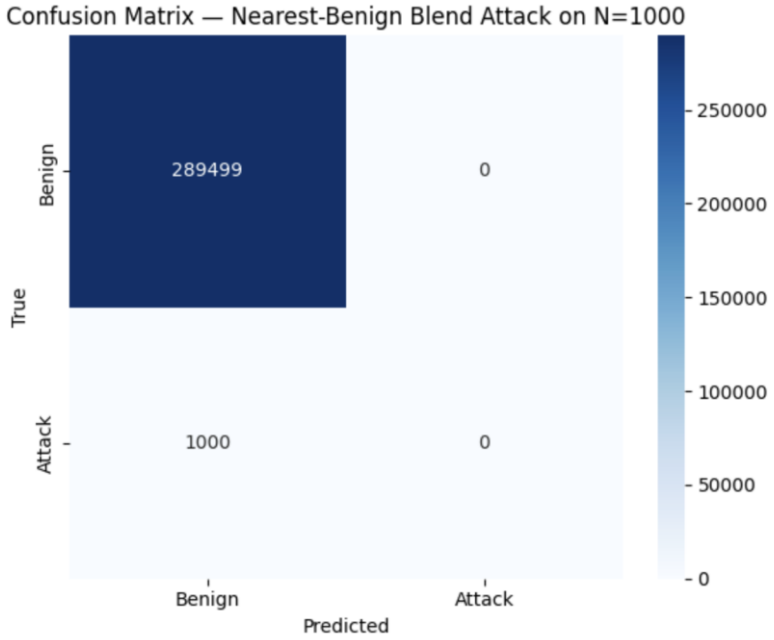


Figure 19. Confusion Matrix – Final Blackbox Blend Attack

5. Conclusions and Future Work

5.1 Summary of Results

This project investigated the performance of machine learning-based network intrusion detection systems (NIDS) when exposed to evasive attacks. Starting from a benchmark dataset (CICIDS2017), both unsupervised and supervised models were developed and evaluated. The focus then shifted to testing these models on real and adversarial traffic generated in a controlled environment.

Through structured whitebox, greybox, and blackbox attack scenarios, the project demonstrated how evasion strategies can degrade detection performance—even in strong baseline classifiers. Finally, adversarial retraining using selected evasive samples

led to a more robust model, capable of detecting stealthier threats while maintaining accuracy on clean traffic.

5.2 Strengths and Weaknesses of the Approaches

Unsupervised (Isolation Forest):

- **Strengths:** Requires no labeled data, fast to train, detects broad anomalies.
- **Weaknesses:** Highly sensitive to shifts in benign behavior; generalized poorly to real traffic; produced many false positives; unable to distinguish subtle malicious patterns.

Supervised (Random Forest):

- **Strengths:** High precision and recall on labeled data; robust to variation in traffic patterns; adaptable through retraining; good feature interpretability.
- **Weaknesses:** Requires labeled data and representative attack samples; initially vulnerable to adversarial manipulation; performance depends heavily on feature quality and coverage.

The supervised model clearly outperformed the unsupervised one, particularly after exposure to adversarial examples. However, it remained susceptible to well-crafted blackbox evasion, highlighting the need for continuous updates and monitoring in any practical deployment.

5.3 Future Directions and Recommendations

This project can serve as a solid foundation for future iterations in this course or related research. Students in following years could:

- **Expand the attack library** by implementing new evasion strategies (e.g., protocol-level abuse, traffic fragmentation, encrypted payloads).
- **Enhance the dataset** with more real-world traffic, multiple victims, and time-based behavior analysis.
- **Compare additional models**, such as deep learning architectures (e.g., LSTM, CNN, Autoencoders), anomaly detection ensembles, or hybrid systems combining multiple detection layers.
- **Explore real-time detection**, integrating models into live monitoring tools or packet capture pipelines.
- **Evaluate long-term robustness**, simulating adaptive attackers using online learning or reinforcement-based evasion.

Improving the detection of evasive attacks remains an open and evolving challenge. This work establishes a practical pipeline that balances academic rigor with real-world applicability—providing a flexible platform for future experimentation and learning.