BUAN 6341 APPLIED MACHINE LEARNING

GROUP PROJECT REPORT

GROUP NO: 4

GROUP MEMBERS: Tara Canugovi, Shriya Reddy Kolan, Vidhya Bharati Kulkarni, Mehul Singh Rajaputhra, Riddhima Reddy Ramasahayam

1. Executive Summary

As cyber threats become more numerous and sophisticated, traditional rule-based Intrusion Detection Systems fail to keep up. Our project provides a hybrid Intrusion Detection System (IDS) with supervised and unsupervised machine learning approaches to effective threat detection. Working on the CIC-IDS2017 dataset that replicates real-world enterprise network traffic, our system has two goals: Detect known threats via supervised models. Detect unknown anomalies via unsupervised methods.

We trained Logistic Regression, Random Forest, XGBoost, and Deep Neural Networks for classification. For novel threat identification, we used Isolation Forest, One-Class SVM, and Autoencoders to learn normality and flag abnormalities. After extensive preprocessing—feature engineering, scaling, encoding, and stratified splitting—Random Forest provided the best precision-recall trade-off for well-known threats. Autoencoders worked incredibly in the detection of new intrusions. SHAP explainability techniques identified the best features, giving security analysts better insight. Together, these models form a layered IDS that improves real-time detection and reduces cyber risk.

2. Introduction

Business Context

Modern cyberattacks often mimic normal traffic, bypassing traditional signature-based systems. Manual monitoring of millions of daily network sessions is infeasible. Machine learning enables automated, adaptive, and scalable threat detection. Machine learning offers a scalable solution to:

- Learn threat patterns from past data
- Detect anomalies in real time
- Adapt to new threats without predefined rules

Objective – Why it's important

This project aims to build a machine learning-based Intrusion Detection System (IDS) that mimics a Security Operations Center (SOC) using session-level network metadata. It combines:

- Supervised Learning for detecting known threats using labeled data.
- Unsupervised Learning to flag anomalies and unknown attacks using unlabeled data.

Together, they provide a robust IDS that automates known threat detection and adapts to new threats.

Relevance

Modern attacks often evade detection. A machine learning IDS minimizes attacker dwell time, reduces damage, and aids compliance with regulations like GDPR, HIPAA, and CCPA.

Data Source

The CIC-IDS2017 dataset (Canadian Institute for Cybersecurity) simulates real enterprise traffic.

3. Data

Data Summary

We worked with a session-level cybersecurity dataset containing metadata from 9,537 network sessions. Each row in the dataset represents a single network session, capturing various characteristics relevant to network activity, user behavior, and potential intrusion indicators.

Dataset Dimensions

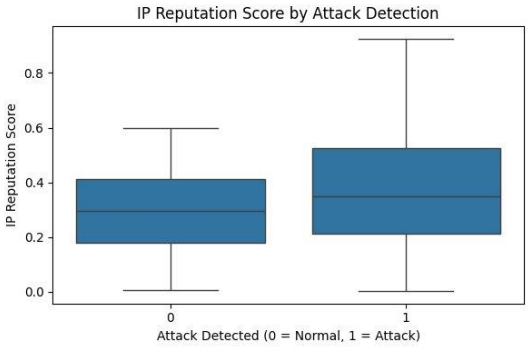
- Total Records: 9,537 sessions
- Total Features: 11 columns

Columns Description

Feature	Type	Description
session_id	Object	Unique identifier for each session (not used for modeling)
network_packet_size	Integer	Total size (in bytes) of packets exchanged during the session
protocol_type	Object	Type of network protocol used (e.g., TCP, UDP)
login_attempts	Integer	Number of login attempts made in the session
session_duration	Float	Duration of the session (in seconds)
encryption_used	Object	Type of encryption applied (e.g., AES, DES, or None)
ip_reputation_score	Float	Trustworthiness of the IP address (higher score indicates a safer connection)
failed_logins	Integer	Count of failed login attempts
browser_type	Object	Browser or user agent used during the session
unusual_time_access	Integer	Flag (0 or 1) indicating access during unusual hours
attack_detected	Integer	Target variable: 1 if an intrusion was detected, 0 otherwise

Exploratory Data Visualization

IP Reputation Score vs. Attack Detection



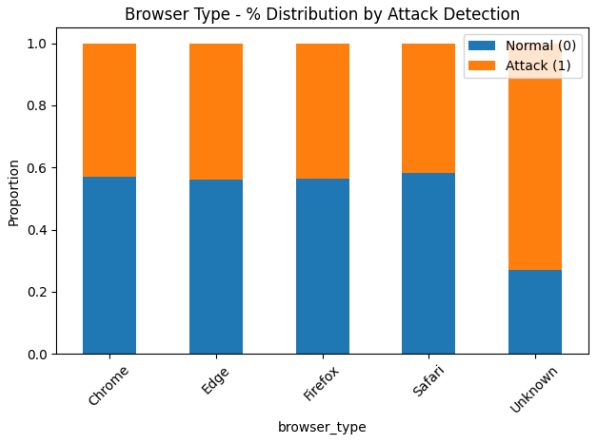
Key Observations

- Attack sessions span a broader IP reputation range (~0.01–0.90+) than normal ones (0.10–0.60).
- High variability suggests use of botnets or spoofed IPs.
- On average, attack sessions have higher reputation scores.

Interpretation

- IP reputation is a valuable signal but not definitive. Its wide variation in attacks underscores the need for multi-feature models to detect evasive threats.

Browser Type Distribution by Attack Detection



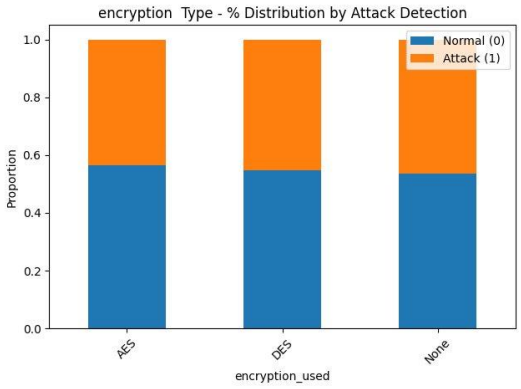
Key Observations:

- Common browsers (Chrome, Edge, etc.) show a balanced mix of normal and attack sessions.
- “Unknown” browsers are ~70% malicious, indicating a strong risk signal.

Interpretation:

- The "Unknown" browser label is often the result of header stripping or spoofing techniques used by attackers to evade detection systems.

Encryption Type vs. Attack Detection



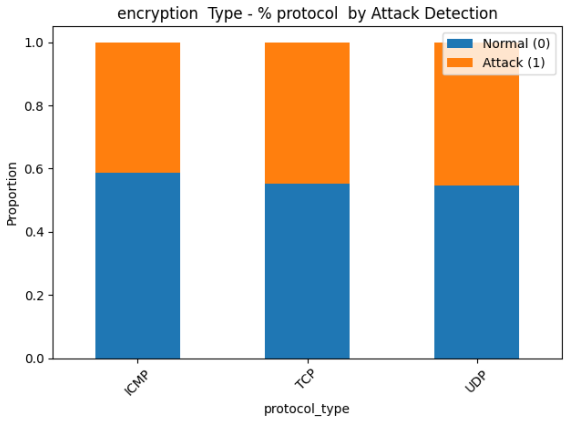
Key Observations:

- AES, DES, and None show similar attack-to-normal ratios.
- No encryption type clearly indicates higher risk.

Interpretation:

- Attackers use encryption just like legitimate users to avoid detection. While essential for privacy, encryption type alone isn’t a strong predictor of malicious behavior but may aid in multi-feature models

Protocol Type vs. Attack Detection

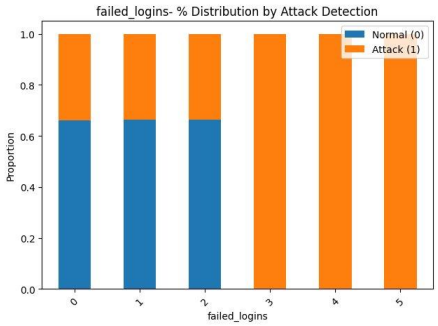


Key Observations:

- Attack and normal session ratios are similar across ICMP, TCP, and UDP.
- No protocol shows a clear preference for attacks.

Interpretation:

- Protocol type alone isn't a strong indicator of malicious activity in this dataset. However, it may still add value when used alongside other features in detection models.



Failed Logins vs. Attack Detection

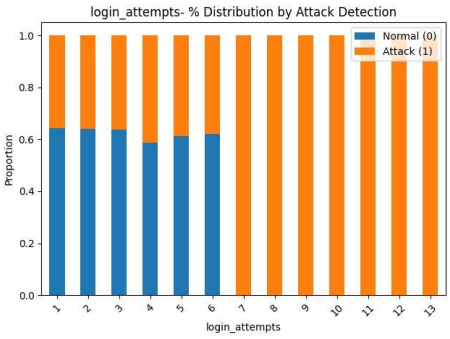
Key Observations:

- 0–2 failed logins are mostly normal.
- 3+ failed logins sharply increase attack likelihood.
- 4–5 failures are almost always malicious.

Interpretation:

- Repeated login failures strongly indicate brute-force or credential attacks, making failed_logins a valuable predictor for intrusion detection.

Login Attempts vs. Attack Detection



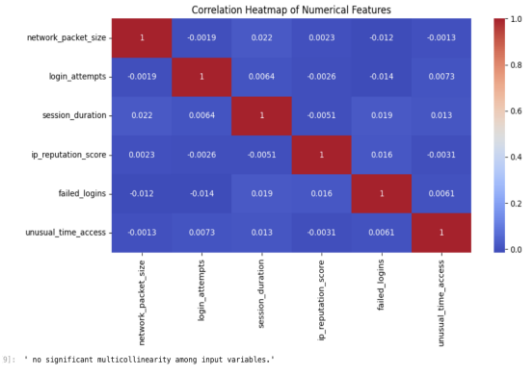
Key Observations:

- 1–6 login attempts show mixed behavior.
- 7+ attempts are mostly attacks, rising steadily through 13.
- Indicates likely brute-force activity.

Interpretation:

- High login attempts strongly signal automated attacks, making login_attempts a key feature for both rule-based and ML-based detection.

Correlation Analysis of Numerical Features



Key Observations:

- No strong correlations found between numerical features.
- Weakest link: session_duration vs. login_attempts ($r \approx 0.0064$).
- All features appear largely independent.

Interpretation:

- Minimal correlation means no multicollinearity—features can be retained for modeling, enhancing both stability and interpretability.

Insights from Exploratory data analysis :

Feature Name	Purpose	Why It's Good
excessive_login_attempts	Flag brute-force attempts with 7 or more login attempts	Matches EDA; simplifies brute-force detection; improves model interpretability
high_ip_reputation_risk	Flag sessions from IPs with high-risk reputation scores	Tunable risk threshold; helps highlight IP-based threats; works well with tree-based models
login_failure_ratio	Capture % of failed login attempts regardless of count	Captures subtle attacks; adds nuance beyond absolute numbers; effective for low-volume threats
excessive_failed_logins	Flag sessions with 3 or more failed login attempts	Mirrors SOC detection rules; strong indicator of credential abuse
is_unknown_browser	Identify automated tools or crawlers using non-standard user agents	Flags suspicious/masked clients; simplifies noisy categorical variable into a strong binary signal

4. Analysis

Analytical Objective

The objective of our analysis is twofold:

- Classify known threats using supervised learning.
- Detect novel or previously unseen threats using unsupervised anomaly detection.

This hybrid strategy ensures both high accuracy on known attack patterns and adaptive detection of emerging threats in real-time network environments.

Logistic Regression

Logistic Regression is a baseline linear model used for binary classification. It provides interpretable coefficients and serves as a benchmark to compare the effectiveness of more complex models like Random Forest.

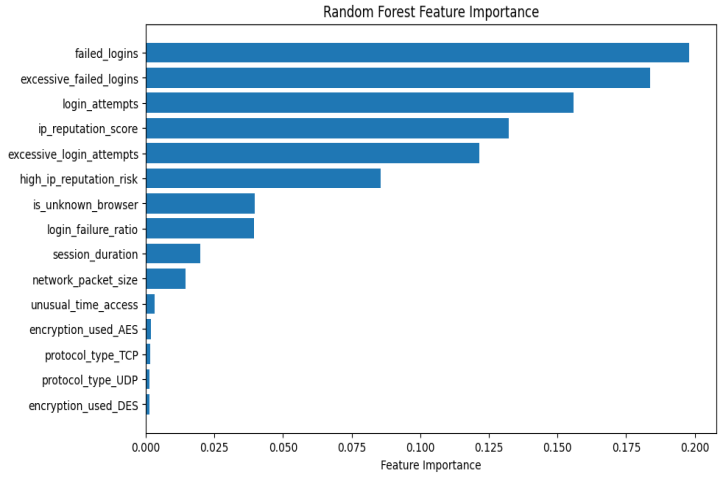
Coefficient Analysis

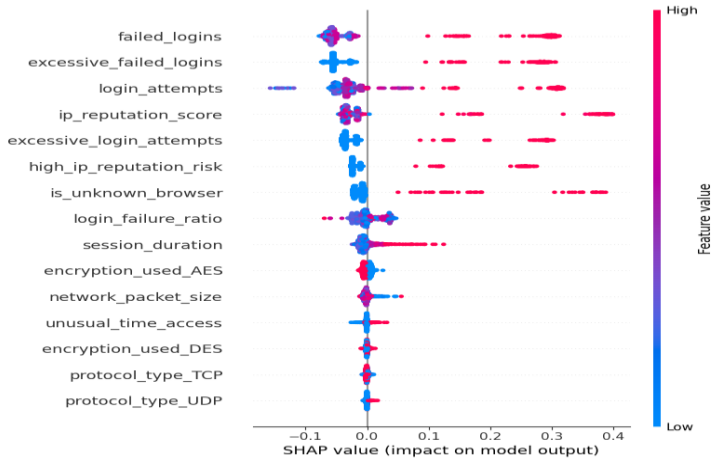
Feature	Coefficient	p-value	95% Confidence Interval
Intercept (const)	-2.0439	0.000	[-2.453, -1.635]
network_packet_size	-0.0167	0.651	[-0.089, 0.056]
login_attempts	0.1090	0.003	[0.038, 0.180]
session_duration	0.1419	0.000	[0.070, 0.213]
ip_reputation_score	0.0118	0.791	[-0.075, 0.099]
failed_logins	-0.0013	0.987	[-0.152, 0.150]
unusual_time_access	0.1723	0.089	[-0.027, 0.371]
excessive_login_attempts	20.6227	0.975	[-1267.407, 1308.653]
high_ip_reputation_risk	22.7693	0.990	[-3511.815, 3557.354]
login_failure_ratio	0.0125	0.894	[-0.172, 0.197]

excessive_failed_logins	25.2493	0.995	[-8227.162, 8277.660]
is_unknown_browser	2.0905	0.000	[1.837, 2.344]
protocol_type_TCP	0.0868	0.605	[-0.242, 0.415]
protocol_type_UDP	0.1748	0.323	[-0.172, 0.521]
encryption_used_AES	-0.2318	0.016	[-0.420, -0.044]
encryption_used_DES	-0.0504	0.625	[-0.253, 0.152]

Random Forest

- Handles Non-linearity: Captures complex interactions that linear models miss.
- Robust to Outliers and Multicollinearity: Makes no strong assumptions about feature distributions.
- Feature Ranking: Provides global feature importance.
- Explainability: When combined with SHAP, it provides local and global interpretability





Key Insights from Random Forest:

Feature	Interpretation
failed_logins	Failed login attempts are the strongest attack signal.
excessive_failed_logins	Repeated failures amplify attack likelihood.
login_attempts	Brute-force behavior is a key indicator.
ip_reputation_score	Risky IPs strongly influence classification.
excessive_login_attempts	Extreme login behavior matters.
high_ip_reputation_risk	Reinforces IP risk as a critical factor.
is_unknown_browser	Browser identity manipulation remains suspicious.
Other Features	Session duration, encryption, and protocol types have minor influence on predictions.

Comparison

Feature	EDA	Logistic Regression	Random Forest (Importance + SHAP)
Failed Logins	Strong EDA pattern	Not significant	Top-ranked, strong SHAP impact
Login Attempts	Strong EDA pattern	Significant (p = 0.003)	Top-ranked, strong SHAP impact
Unknown Browser	~70% attacks	Strongest predictor (p = 0.000)	Moderate importance in RF + SHAP
IP Reputation	High risk in EDA	Not significant	High importance in RF + SHAP
Session Duration	Mild EDA signal	Significant (p = 0.000)	Low importance in RF + SHAP
AES Encryption	No clear trend	Protective (p = 0.016)	Minimal importance

Conclusion

- Consistent Signals:
  - Login behaviors (failed logins, excessive attempts) emerge as strongest predictors across all methods.
- Model-Specific Strengths:
  - Logistic Regression provides statistical significance for login_attempts and unknown_browser.
  - Random Forest captures complex behaviors, especially failed login patterns and IP risk, missed by logistic regression.
- SHAP Explains Why:
  - SHAP shows how high and low values of features shift predictions toward attack or normal, providing actionable transparency.
- Business Recommendation:
  - Deploy Random Forest with SHAP monitoring for high explainability and operational effectiveness.

5. Results

a) Baseline Benchmarks

We trained Logistic Regression and Random Forest models on the raw dataset without applying scaling, advanced encoding, or feature engineering. The goal was to establish a baseline to compare the impact of further pre-processing and tuning.

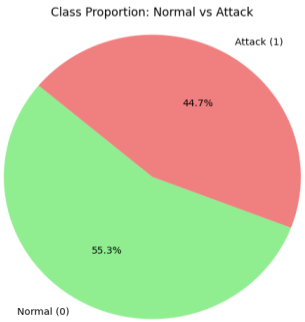
i) Baseline models without Pre-preprocessing

Metric	Logistic Regression	Random Forest
Accuracy	73%	89%
Precision	71%	100%
Recall	66%	75%
F1-Score	68%	85%
False Positives	234 (Normal classified as Attack)	0
False Negatives	289 (Missed real attacks)	217 (Missed real attacks)
Insight	Struggles without preprocessing. High false positives and false negatives. Risky for operational use without tuning.	High precision but misses 25% of attacks. Needs tuning to balance precision and recall.

Comparative Takeaway

- Logistic Regression is too error-prone without preprocessing.
- Random Forest is highly precise but recall-limited, potentially dangerous if attackers exploit what it fails to detect.
- Both models require preprocessing and threshold optimization to balance recall and precision effectively.

ii) Report class distribution



Why the Class Imbalance Exists

In real-world network traffic, attacks are rare (<1%), but this dataset contains 45% attacks due to intentional oversampling. This was done to give the model enough attack examples to learn from and avoid bias toward predicting only "Normal" traffic.

Impact

- Positive: Improves learning and sensitivity during model training.
- Negative: May overestimate real-world performance, as actual attack rates are much lower, making precision and recall less realistic in this balanced setting.

b) Pre-processing Steps Applied

- Handled Missing Values: Filled missing encryption_used with 'None'.
- Removed Non-Predictive Columns: Dropped session_id.
- One-Hot Encoding: Converted categorical features.

- Feature Engineering: Added behavioral flags (e.g., excessive login attempts).
- Stratified Train-Test Split: Balanced class representation.

Model performance changes after pre-processing

Metric (Class 1 - Attack)	Logistic Regression	Random Forest	XGBoost
Accuracy	88%	88.5%	88.4%
Precision	98.3%	100%	100%
Recall	74.2%	74.3%	74.1%
F1-Score	84.6%	85.3%	85.1%

Which Metrics Matter Most and Why

- Primary Focus – Recall (Sensitivity) for Attack Class (1)
  - Why? In cybersecurity, missing attacks is riskier than false alarms.
  - Recall tells us how many real attacks the model actually catches.
  - Models currently have ~74% recall, meaning 1 in 4 attacks go undetected, which is still a concern.
- Secondary Focus – Precision for Attack Class (1)
  - Why? We don't want to overwhelm security analysts with false alarms.
  - Models achieved 98-100% precision, meaning they rarely raise false alarms—which is excellent.
- Balanced Metric – F1-Score for Attack Class (1)
  - Why? F1 balances both false alarms and missed attacks.
  - Models have 85%+ F1-Score, showing they handle both reasonably well.
- Accuracy is Less Relevant
  - Why? High accuracy can be misleading in imbalanced scenarios, since most sessions are normal in real life.

How Pre-processing Affected the Metrics

Metric	Before Pre-processing	After Pre-processing	Impact
Accuracy	73% (LR) / 89% (RF)	88% (LR) / 89% (RF/XGB)	Improved overall correctness, especially for Logistic Regression.
Precision	71% (LR) / 100% (RF)	98% (LR) / 100% (RF/XGB)	Significantly reduced false alarms for Logistic Regression while keeping RF perfect.
Recall	66% (LR) / 75% (RF)	74% (LR) / 74% (RF/XGB)	Slightly improved attack detection for Logistic Regression, no change for RF/XGB.
F1-Score	68% (LR) / 85% (RF)	85% (LR) / 85% (RF/XGB)	Balanced improvement, especially for Logistic Regression.

Summary of Pre-processing Benefits

- Made Logistic Regression usable by improving both recall and precision.
- Helped all models generalize better without overfitting.

- Reduced unnecessary false alarms while slightly improving detection rates.

(c) Iterative Improvement

Experiments Conducted

- Behavioral Feature Engineering: Captured attacker behaviors like brute-force login attempts.
- Feature Selection: Dropped low-impact features based on SHAP and importance.
- SMOTE Resampling: Balanced training data to improve attack detection.
- Unsupervised Anomaly Detection: Evaluated Isolation Forest, One-Class SVM, and Autoencoder.

Comparison to Baseline Performance

Method	Precision	Recall	F1-Score	Accuracy
Supervised Baseline (RF)	100%	74%	85%	89%
After EDA (RF)	98%	75%	85%	88%
After SMOTE Resampling (RF)	96%	77%	85%	88%
Isolation Forest (Unsupervised)	56%	31%	40%	58%
One-Class SVM (Unsupervised)	76%	79%	77%	79%
Autoencoder (Unsupervised)	86%	79%	82%	85%

Autoencoder provided the best balance for detecting unknown attacks without labels, complementing supervised models which focus on known threats.

Why Use Unsupervised Anomaly Detection

- Real-world attacks constantly evolve, often lacking labeled data.
- Unsupervised models learn normal behavior and flag deviations as potential threats, making them ideal for detecting rare, unseen, or zero-day attacks.

6. Discussion

a) Key Learnings

- Supervised Models (Random Forest) performed best on known attacks with high precision (98–100%), but missed 1 in 4 attacks (74–77% recall), posing operational risk.
- Feature Engineering and SMOTE slightly improved recall without sacrificing precision, but missed attacks remain a concern.
- Unsupervised Anomaly Detection (Autoencoder) achieved 86% precision and 79% recall, making it better suited for detecting unknown or novel threats that supervised models cannot recognize.

b) Business Implications

- Hybrid detection reduces the risk of both known and unknown attacks.
- High precision minimizes analyst overload, improving response efficiency.
- Explainable models support regulatory compliance (GDPR, HIPAA).
- Models are deployment-ready for SIEM integration and real-time monitoring.

c) Recommendations

- Deploy a Hybrid System: Random Forest for known attacks, Autoencoder for unknown threats.
- Continuously Update Models: Retrain with new network data to stay ahead of attackers.
- Collect Better Data with Honeypots: Generate high-quality attack data for future improvements.
- Time-series models for session tracking. Federated learning for secure cross-organization knowledge sharing.