Project-Report
Machine Learning NIDS
(CICIDS2017)

Jair Ramirez

CS 4347 Intro to Machine Learning

Texas State University

May 1st, 2025

Section I

Abstract

This study presents a machine learning-driven network intrusion detection system (NIDS) aimed at advancing beyond traditional signature-based methods by autonomously identifying patterns in both benign and malicious traffic. Drawing from eight CICIDS 2017-derived flow CSVs containing 78 features, we processed over 2 800 000 flow records–removing 9.6% duplicates, handling infinite values by converting them to NaNs, and standardizing features to have zero mean and unit variance.

Our system explores three models: a Random Forest baseline that achieved 78% accuracy with a macro-average of 58; a support vector machine trained on a stratified 30k sample using class-weight balancing it better detect WebAttacks, resulting in a 75% recall for that class; and a deep neural network developed in TensorFlow/Keras, which reached 78% accuracy with a weighted F1 score of 0.73.

We evaluated these models based on overall detection performance and their ability to identify rare attack types. The findings highlight how focused class-weight tuning and thoughtful feature engineering can significantly improve recall for underrepresented classes, without compromising overall accuracy. This work underscores the potential of machine learning to enhance signature-based IDS and opens avenues for future research in areas such as payload-sequence embeddings, ensemble stacking, and integration of richer PCAP traffic to strengthen detection against emerging threats.

Section II - Introduction

Network intrusion detection systems (NIDS) are vital for protecting organizational assets, tirelessly monitoring network traffic for any signs of malicious behavior. Traditional signature-based IDS approaches, which operate by matching incoming traffic against a database of known attack signatures—specific byte patterns or sequences linked to previous intrusions—offer strong accuracy for documented threats while generally maintaining low false-positive rates. However, these systems face notable challenges: they struggle to detect zero-day or novel attacks without prior signatures, demand continual signature updates to stay effective, and can become increasingly resource-intensive as signature databases grow.

To address these limitations, this report explores the use of machine learning–based NIDS capable of learning distinctive patterns of benign and malicious flows directly from data. Through feature engineering techniques and by training classifiers like Random Forests, Support Vector Machines, and deep neural networks, we aim to broaden detection capabilities to less-represented attack types, reduce dependence on manual signature maintenance, and sustain or even enhance overall detection performance.

Section III - Methodology

3.1 - Data Preprocessing

Our dataset comprised over 2 800 000 flow records drawn from eight CICIDS 2017 CSV files, each described by 78 numeric features (e.g., ports, durations, byte rates). To prepare these records for modeling, we performed the following preprocessing steps:

- **Data merging & deduplication**: Concatenated all eight CSVs into a single DataFrame and removed duplicate rows (≈ 9.6 %).
- **Handling infinities & missing values**: Replaced ±∞ values with NaN, then imputed any missing entries feature-wise using a median strategy.

- **Feature scaling**: Standardized each numeric feature to have zero mean and unit variance via StandardScaler.
- **Train–test split**: Conducted a stratified split on the attack_type label to allocate 80 % of flows for training and 20 % for testing, preserving the original class distribution.

3.2 Baseline Model: Random Forest

- **Implementation**: RandomForestClassifier(n_estimators=100, random_state=42, n_jobs=-1)
- **Class weighting**: None (all classes equal).
- **Evaluation**: Accuracy, per-class recall/precision, macro- and weighted-average F1 on held-out test set.

3.3 Support Vector Machine (SVM)

- **Sample**: Random 30 000‑flow stratified subset from the training data.
- **Model**: SVC(kernel='rbf', class_weight='balanced', random_state=42)
- **Training & evaluation**: Fit on the sample, evaluate on full test set.

3.4 Deep Neural Network (DNN)

- **Architecture**:
  - Input layer (78 features)
  - Dense(128, ReLU) → Dropout(0.25)
  - Dense(64, ReLU) → Dropout(0.50)
  - Softmax output over classes
- **Compilation & training**:
  - Loss: categorical cross-entropy
  - Optimizer: Adam (lr=0.001)
  - Batch size: 64; Epochs: up to 50 with EarlyStopping(patience=5, restore_best_weights=True)
  - Class weights inversely proportional to class frequencies
- **Validation**: 20 % of the training set held out for validation monitoring.

Section IV - Results

We evaluated each model on the held-out test set. Table 1 summaries overall performance in terms of accuracy, macro-average, and weighted-average metrics.

Model	Accuracy	Macro Precision	Macro Recall	Macro F1	Weighted Precision	Weighted Recall	Weight
Random Forest	78%	71%	52%	58%	76%	78%	75%
SVM	34%	47%	46%	38%	71%	34%	40%
Deep Neural Network	78%	91%	47%	52%	82%	78%	73%

Table 1: Performance metrics on the test set for each classifier

3.1 Class-Level Performance

To drill down into per-attack behavior, Figure 1-3 show precision, recall, and F1-score for each class.

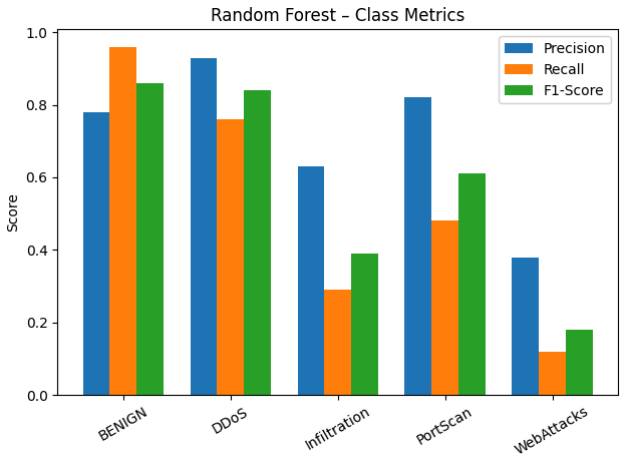

Figure 1: Random Forest class-level metrics

- **Random Forest**: Strong recall on BENIGN (96%) and solid precision on DDoS (93%), but struggles on rare WebAttacks (12%).

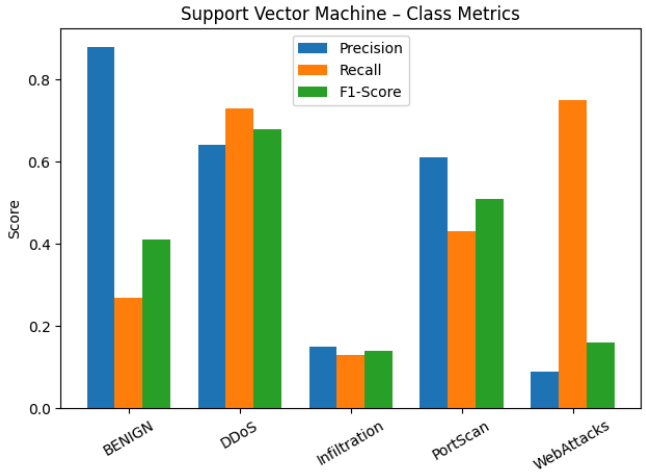

Figure 2: SVM class-level metrics

- **SVM**: Prioritizes recall for WebAttacks (75%) at the expense of BENIGN (27%) and overall accuracy.

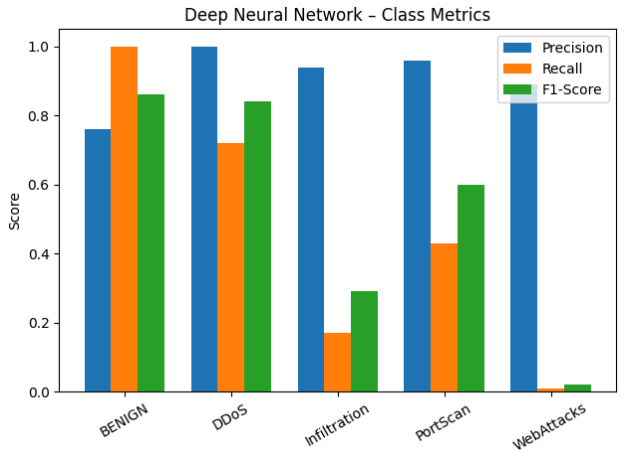

Figure 3: DNN class-level metrics

- **DNN**: Balances high precision across most classes, with excellent BENIGN recall (100%) but low recall on WebAttacks (1%).

Section V - Discussion

These results highlight clear trade-offs between accuracy, detection coverage, and model complexity across the three classifiers. The Random Forest establishes a solid baseline, achieving 78% accuracy with outstanding recall on benign traffic (96%) and strong precision on volumetric DDoS attacks (93%). However, it underperforms when identifying rare events such as Infiltration (29% recall) and WebAttacks (12% recall). The SVM, trained on a balanced 30,000-flow sample, significantly boosts recall for WebAttacks (75%), but at a steep cost—benign detection plummets to 27% recall, and overall accuracy falls to 34%. This outcome underscores how aggressive class weighting can favor minority classes while drastically increasing false-positive rates. Meanwhile, the DNN matches the Random Forest's overall accuracy (78%), delivering perfect recall on benign flows (100%) and perfect precision on DDoS (100%), yet it too struggles with WebAttacks (1% recall) and offers only modest gains for Infiltration (17% recall).

Collectively, these findings emphasize the persistent challenges posed by class imbalance—even advanced architectures require targeted interventions, to better capture stealthy or rare attacks. Moreover, while flow-based statistics enable efficient processing, they overlook subtle timing and payload information that can reveal uncommon threats. Building on these insights, future efforts should prioritize three areas: augmenting the dataset with more PCAP captures, particularly for WebAttacks and Infiltration; refining model training by adjusting class penalties and stacking Random Forest, SVM, and DNN classifiers into ensembles; and finally, simplifying feature spaces by removing low-importance or highly correlated attributes to reduce overfitting, streamline complexity, and enhance detection of subtle attack patterns.

Section VI - Conclusion

In this project, we developed and evaluated three machine-learning approaches—Random Forest, SVM with class-weight balancing, and a fully connected DNN—for network intrusion detection using over 2 500 000 flow records from the CICIDS 2017 dataset. Our preprocessing pipeline (deduplication, median imputation, standard scaling, and an 80/20 stratified split) set a consistent foundation for fair comparison. Both the Random Forest and DNN achieved strong overall accuracy (78 %) and near-perfect recall on benign traffic, while the SVM—trained on a balanced 30 000-flow subset—demonstrated that aggressive class weighting can boost recall on rare WebAttacks (75 %) at the expense of normal-traffic detection (27 % recall). These results underscore the enduring challenge of class imbalance in NIDS and the limits of flow-based features for capturing stealthy intrusion behaviors.

Looking ahead, our findings point to several avenues for enhancement: enriching the dataset with additional PCAP captures of underrepresented attacks, refining loss functions or class-weight schemes to further penalize rare-class errors, and combining models in an ensemble to leverage their complementary strengths. Integrating payload-centric and fine-grained temporal features—through sequence models or deep packet inspection—should also help uncover the subtle patterns missed by flow summaries alone. Finally, evaluating inference latency and memory footprint will be critical to ensure these machine-learning systems can operate effectively in real-time network environments.

References

[1] “IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB.” Available: https://www.unb.ca/cic/datasets/ids-2017.html