Article

Not peer-reviewed version

A Deterministic Comparison of Classical Machine Learning and Hybrid Deep Representation Models for Intrusion Detection on NSL-KDD and CICIDS2017

Miguel Arcos-Argudo [*,†,‡], Rodolfo Bojorque [‡], Andrés Torres

Posted Date: 7 November 2025

doi: 10.20944/preprints202511.0425.v1

Keywords: intrusion detection system (IDS); NSL-KDD; CICIDS2017; Naïve Bayes; logistic regression; linear discriminant analysis; autoencoder; SMOTE; AUC; false alarm rate

Article

A Deterministic Comparison of Classical Machine Learning and Hybrid Deep Representation Models for Intrusion Detection on NSL-KDD and CICIDS2017

Miguel Arcos-Argudo [1,*,†,‡], Rodolfo Bojorque [1,‡] and Andrés Torres [2]

[1] Math Innovation Group, Universidad Politécnica Salesiana, Cuenca 010102, Ecuador
[2] Universidad Católica de Cuenca, Cuenca 010101, Ecuador
* Correspondence: marcos@ups.edu.ec
† Current address: Math Innovation Group, Cuenca Campus, Universidad Politécnica Salesiana - Turuhuayco Ave. 3-69, Cuenca 010210, Ecuador.
‡ These authors contributed equally to this work.

Abstract

Intrusion detection systems (IDS) must balance detection quality with operational transparency. We present a *deterministic, leakage-free* comparison of three classical classifiers: Naïve Bayes, Logistic Regression, and Linear Discriminant Analysis; and we propose a hybrid pipeline that trains LR on autoencoder embeddings. Experiments use NSL-KDD and CICIDS2017 under two regimes (with/without SMOTE applied *only* on training data). All preprocessing (one-hot encoding, scaling, and imputation) is fit on the training split; fixed seeds and deterministic TensorFlow settings ensure exact reproducibility. We report a *complete* metric set—Accuracy, Precision, Recall, F1, AUC, and False Alarm Rate (FAR)—and release a replication package (code, preprocessing artifacts, and saved prediction scores) to regenerate all reported tables and metrics. On NSL-KDD, AE+LR yields the highest AUC (≈ 0.904) and the strongest F1 among the evaluated models (e.g., 0.7583 with SMOTE), while LDA slightly edges LR on Accuracy/F1. NB attains very high Precision (≈ 0.98) but low Recall (≈ 0.24), resulting in the weakest F1 yet a low FAR due to conservative decisions. On CICIDS2017, LR delivers the best Accuracy/F1 (0.9878/0.9752 without SMOTE), with AE+LR close behind; both approach ceiling AUC (≈ 0.996). SMOTE provides modest gains on NSL-KDD and limited benefits on CICIDS2017. Overall, LR/LDA remain strong, interpretable baselines, while AE+LR improves separability (AUC) without sacrificing a simple, auditable decision layer for practical IDS deployment.

Keywords: intrusion detection system (IDS); NSL-KDD; CICIDS2017; Naïve Bayes; logistic regression; linear discriminant analysis; autoencoder; SMOTE; AUC; false alarm rate

1. Introduction

In today's hyperconnected society, safeguarding digital infrastructures has become a cornerstone of national, economic, and social stability. As cyber threats continue to evolve in scale and complexity, organizations must adopt intelligent and proactive defense mechanisms. Intrusion Detection Systems (IDS) represent a critical layer in this defense architecture, serving to monitor and identify malicious or anomalous activity in real time. Traditionally, IDS have relied on signature-based or rule-based techniques, which match observed patterns against known attack signatures. Although effective against previously encountered threats, these approaches often fail to detect new or obfuscated attacks and typically suffer from high false positive rates [1]. To address these limitations, recent efforts have turned to machine learning (ML), deep learning (DL), and hybrid methods capable of generalizing from data and adapting to new threat landscapes.

Machine learning models (both supervised and unsupervised) enable systems to learn behavioral patterns from historical traffic data [2]. Classical classifiers: Naïve Bayes (NB), Logistic Regression (LR),

 doi:10.20944/preprints202511.0425.v1

and Linear Discriminant Analysis (LDA) apply well-established mathematical frameworks to detect anomalies in network activity [1]. While these models are often interpretable and computationally lightweight, they can struggle with nonlinear relationships and high-dimensional feature spaces typical of modern cybersecurity datasets.

In contrast, deep learning (DL) architectures provide powerful tools for automated representation learning and pattern recognition. In particular, autoencoders (AE) can learn compact, informative latent embeddings from network traffic data [4–6]. These embeddings can then be fed to simple yet effective downstream classifiers such as LR, yielding hybrid pipelines (AE+LR) that combine unsupervised feature learning with interpretable decision boundaries. Such DL-based representations have demonstrated strong performance in identifying complex and evolving cyber threats [1,7]. However, practical deployment still faces challenges related to computational demands, data requirements, and explainability [8,9].

In the context of applied AI and cybersecurity engineering, conducting rigorous comparative studies is vital. Such evaluations clarify when interpretable statistical models are sufficient for effective detection and when representation learning can provide measurable gains when paired with lightweight classifiers.

This study presents a deterministic comparison of three classical classifiers (NB, LR, and LDA) and a hybrid deep-learning pipeline (AE+LR) for intrusion detection [10]. To ensure robustness, we employ two widely used benchmark datasets: NSL-KDD [11] (derived from KDD'99) and CICIDS2017 [12], which include up-to-date traffic and attack types. These datasets provide a balanced basis to examine model generalization, performance across diverse attacks, and real-world feasibility.

We assess each model using standard performance metrics: accuracy, precision, recall, F1-score, AUC, and FAR. In addition, we provide complete tabular reporting. The key contribution of this paper include:

(i) a *unified, leakage-free, and deterministic* evaluation protocol applied consistently across NB, LR, LDA, and a hybrid AE+LR pipeline; (ii) dual training regimes (with/without SMOTE strictly on training folds) to isolate the effect of class rebalancing; (iii) a transparent *deep-representation + linear classifier* baseline (AE+LR) that combines competitive AUC/F1 with auditability; (iv) complete tables (Accuracy, Precision, Recall, F1, AUC, FAR) for both NSL-KDD and CICIDS2017; and (v) a replication package (code, configs, fixed seeds) enabling reproducibility.

The remainder of this article is structured as follows. Section 2 provides a review of related research on IDS using ML and DL. Section 3 describes the experimental methodology. Section 4 presents the results. Section 5 discusses the findings, limitations, and contextual relevance. Section 6 concludes with recommendations and directions for future work.

2. Evolution of Machine Learning and Deep Learning for IDS

2.1. Classical ML Baselines for IDS

Early applications of machine learning to intrusion detection focused on algorithms like decision trees, support vector machines, k-nearest neighbors, Naïve Bayes, and logistic regression to classify network traffic as normal or attack [13]. For example, Gu and Lu [14] proposed an SVM-based IDS enhanced with Naïve Bayes feature embedding, which improved detection performance by leveraging probabilistic feature–class relationships. Traditional ML models, however, often encounter challenges with the high dimensionality and complexity of network data. Naïve Bayes (NB) assumes feature independence and can underperform when features are correlated (common in network traffic), leading to high false positives especially on minority attack classes [1]. Logistic Regression (LR) is a linear classifier that tends to perform well only if the data is roughly linearly separable [1]. LDA (Linear Discriminant Analysis), similarly, finds linear combinations of features that separate classes; it has been used both for dimensionality reduction and classification in IDS studies. For instance, a recent work in an industrial IoT context reported ~97% detection accuracy using a wrapper-based LDA classifier on the NSL-KDD dataset [15]. Overall, while fast and interpretable, these statistical models may struggle

 doi:10.20944/preprints202511.0425.v1

to capture non-linear relations in network traffic features or the complex structures present in modern attack behaviors.

2.2. Deep Learning Approaches and AE-Based Representation Learning

In parallel, deep learning approaches gained popularity for their ability to learn feature representations, beginning with intrusion detection using feedforward neural network (FFNN) classifiers [16,17]. Autoencoders (AE), as unsupervised neural networks, learn to compress data into a lower-dimensional latent space and reconstruct it. They have been actively studied in IDS because they can learn to model typical network traffic and derive informative latent embeddings [3]. Several studies analyze diverse AE architectures for intrusion detection, showing that model capacity and bottleneck size strongly affect downstream performance; for example, a simple stacked AE has reported F1-scores close to 0.90 on NSL-KDD when properly tuned [3]. Other works further improve AE-based pipelines by addressing data biases and outliers, reaching accuracies around 90% on NSL-KDD [18]. While many AE studies use reconstruction-error thresholds for anomaly scoring—whose calibration can be delicate [10]—an alternative is to use AE purely as a *representation learner* and train a lightweight supervised classifier on top of its embeddings (e.g., AE+LR), which removes the need for threshold tuning while retaining interpretability.

2.3. Ensembles and Imbalance Remedies

Beyond single models, ensemble strategies have been explored to boost robustness. Roy et al. [21] introduced a stacking ensemble of boosted decision tree models, evaluated on IoT traffic with NSL-KDD and CICIDS2017; their system obtained a high overall accuracy of 98.5% on NSL-KDD and 99.11% on CICIDS2017, though detection of rare attack classes (U2R, R2L) remained challenging. De Souza et al. [22] reported a two-step ensemble (ExtraTrees and neural networks) that achieved an impressive 99.81% accuracy on NSL-KDD; however, even this advanced model detected only 68.75% of U2R attacks, underscoring the persistent difficulty of heavily imbalanced distributions. To mitigate these issues, recent work frequently incorporates data oversampling, cost-sensitive learning [23], and feature selection.

In summary, the literature indicates that learned representations from deep models—particularly autoencoders—often enhance detection accuracy and adaptability to evolving attacks when paired with simple classifiers, while classical methods (NB, LR, LDA) remain attractive for speed, simplicity, and explainability [1]. Our study builds on these insights by directly comparing representative classical classifiers (NB, LR, LDA) against a hybrid pipeline that uses unsupervised deep representations with a linear decision layer (**AE+LR**) under a unified experimental setup.

3. Methodology

We design a fair, *deterministic* and *leakage-free* evaluation to compare classical machine learning baselines with a hybrid deep-representation pipeline for IDS. All preprocessing/resampling steps are fit *only* on training data; seeds are fixed across Python/NumPy/TensorFlow (42), enabling exact reproducibility.

3.1. Evaluation Protocol and Datasets

We adopt a *held-out* scheme in both benchmarks: for **NSL-KDD** we use the official *KDDTrain+*/*KDDTest+* split; for **CICIDS2017** we use a single 80/20 *stratified* split with `random_state=42`. We evaluate the **binary** setting (attack vs. normal) to focus on primary detection, consistent with prior IDS literature [1,18,21,27].

NSL-KDD (improved KDD'99) contains 41 features plus label and four attack families [11]. **CICIDS2017** captures modern traffic/attacks with ∼80 numerical flow features [12,24]. External validity caveat: NSL-KDD is legacy and CICIDS2017 is controlled; we retain them for comparability and availability.

 doi:10.20944/preprints202511.0425.v1

Table 1. Class distribution in the NSL-KDD dataset (KDDTrain+ and KDDTest+ combined).

Class	Training (#)	Testing (#)	Total
Normal	67,343	9,710	77,053
DoS	45,927	7,458	53,385
Probe	11,656	2,887	14,543
R2L	995	2,421	3,416
U2R	52	67	119
Total	**125,973**	**22,543**	**148,516**

Table 2. Attack categories from CICIDS2017 used in this study (80/20 stratified split; natural class imbalance retained).

Category	Example Attacks	# Samples (train)	# Samples (test)
Brute Force	FTP-Patator, SSH-Patator	5,578	1,394
DoS/DDoS	DoS Hulk, LOIC, HOIC, DDoS	345,376	86,344
Web Attack	XSS, SQL Injection, etc.	1,645	411
Botnet	ARES botnet traffic	3,842	961
Infiltration	Unauthorized access (Infiltration)	57,478	14,370
Heartbleed	Heartbleed exploit	9	2
Benign	Legitimate traffic	1,266,053	316,513
Total	(Benign + all attacks)	**1,679,981**	**419,995**

3.2. Preprocessing (Uniform and Leakage-Free)

Pipelines are implemented uniformly: one-hot encoding for NSL-KDD categorical fields, min–max scaling to $[0, 1]$, and median imputation for CICIDS2017 numerics. All transformers are fit on training only and applied to the held-out test split. To mitigate imbalance, we use **SMOTE** on training data only (no resampling in validation/test), preserving original test distributions—a standard, transparent practice in IDS [27,30].

3.3. Models and Hyperparameters

We compare well-established **baselines** (NB, LR, LDA) recommended by recent surveys for transparency/reproducibility [26,27] against a **hybrid** deep-representation pipeline (AE+LR) supported by prior IDS evidence [26,28,29]. We deliberately use *minimal, uniform* tuning to avoid family-specific advantages.

 doi:10.20944/preprints202511.0425.v1

Table 3. Model configurations (shared across datasets unless noted).

Model	Key setup / hyperparameters
Naïve Bayes (GaussianNB)	Gaussian likelihood; no tunables. Pipeline: (NSL-KDD) One-hot → Min–max → NB; (CICIDS) Imputer(median) → NB. With SMOTE: resampling applied inside training pipeline only.
Logistic Regression (LR)	Linear classifier with L2; `max_iter=1000`. (NSL-KDD) One-hot → Min–max → LR. (CICIDS) Imputer(median) → LR. With SMOTE: Imputer → SMOTE → LR.
Linear Discriminant Analysis (LDA)	Shared-covariance Gaussians; regularized variant recommended for tabular IDS [31]. (NSL-KDD) default LDA on scaled one-hot. (CICIDS) `solver=lsqr`, `shrinkage=auto` after imputation; with SMOTE analogous pipeline.
Autoencoder + LR (AE+LR)	Shallow symmetric AE: Dense(64, ReLU) → Dense(32, ReLU) → Dense(64, ReLU) → Output(sigmoid). Loss: MSE; Adam $\eta = 10^{-3}$; batch 256; 20 epochs; `validation_split=0.1`; `shuffle=False`. Train AE on training inputs only, extract $z = f_{\text{AE}}(x)$; fit LR(`max_iter=1000`) on z_{train}. SMOTE (when used) is applied on embeddings z_{train}.
SMOTE (train only)	`random_state=42`, `k_neighbors=5`, `sampling_strategy='auto'`. No resampling on test.

3.4. Metrics and Reproducibility

We report **Accuracy**, **Precision**, **Recall**, **F1**, **AUC** (from `predict_proba`) and **False Alarm Rate (FAR)**, with

$$\text{FAR} = \frac{FP}{FP + TN}. \tag{1}$$

Deterministic TensorFlow ops are enabled; exact seeds, scripts and stored prediction scores are released in a replication package (code/configs/`predict_proba`), enabling byte-for-byte regeneration of tables/metrics.

All code, preprocessing artifacts, trained models, and configuration files are publicly available in our replication repository at https://github.com/miguelarcosa/IDS-KDD-CICIDS2017.git, ensuring full byte-level reproducibility of all tables and metrics reported in this paper.

4. Results

This section reports the performance of the four evaluated approaches: Naïve Bayes (NB), Logistic Regression (LR), Linear Discriminant Analysis (LDA), and the hybrid Autoencoder embeddings + Logistic Regression (AE+LR) in NSL-KDD and CICIDS2017 datasets. We present Accuracy, Precision, Recall, F1, AUC, and False Alarm Rate (FAR), consistent with the evaluation protocol described earlier. Unless noted, models use default thresholds (0.5 for probabilistic outputs), and all preprocessing/SMOTE steps are fit on the training split only.

4.1. Overall Performance on NSL-KDD

Tables 4 and 5 summarize the binary results on NSL-KDD without and with SMOTE, respectively. Across both regimes, LR, LDA, and AE+LR clearly outperform NB, which shows very high Precision but poor Recall (i.e., conservative attack labeling that misses many positives).

For reference, after applying SMOTE only on the training split, the NSL-KDD training set becomes exactly balanced at Benign:Attack = 80,892 : 80,892 (1:1).

 doi:10.20944/preprints202511.0425.v1

Table 4. Performance metrics of all models on the NSL-KDD test set (*without* SMOTE).

Model	Accuracy	Precision	Recall	F1-score	FAR	AUC
Naïve Bayes	0.5649	0.9800	0.2406	0.3864	0.0065	0.7987
Logistic Regression (LR)	0.7443	0.9142	0.6079	0.7302	0.0754	0.8276
LDA	0.7616	0.9249	0.6326	0.7514	0.0679	0.8484
Autoencoder (AE) + LR	0.7616	0.9165	0.6397	0.7534	0.0770	0.9040

Table 5. Performance metrics of all models on the NSL-KDD test set (*with* SMOTE on training data only).

Model	Accuracy	Precision	Recall	F1-score	FAR	AUC
Naïve Bayes	0.5650	0.9800	0.2408	0.3866	0.0065	0.7978
Logistic Regression (LR)	0.7458	0.9137	0.6112	0.7324	0.0763	0.8227
LDA	0.7632	0.9238	0.6365	0.7537	0.0694	0.8525
Autoencoder (AE) + LR	0.7654	0.9166	0.6467	0.7583	0.0778	0.9043

Key observations.

- *NB*: Exceptionally high Precision (≈0.98) but low Recall (≈0.24) produces the lowest F1; FAR is minimal due to the conservative decision boundary.
- *LR vs. LDA*: LDA edges LR slightly on Accuracy/F1 and AUC in both regimes; differences are modest.
- *AE+LR*: Delivers the highest AUC (≈0.904) and the best Recall/F1 trade-off among the four, with Accuracy comparable to LDA.
- *Effect of SMOTE*: Improvements are small but consistent where they appear (e.g., LDA F1: 0.7514→0.7537; AE+LR F1: 0.7534→0.7583), indicating mild gains in detecting attacks at an approximately unchanged FAR.

4.2. Overall Performance on CICIDS2017

Tables 6 and 7 summarize binary results on CICIDS2017. Performances are generally high across LR, LDA, and AE+LR, with NB trailing due to a high FAR and lower Precision.

Table 6. Performance metrics of all models on the CICIDS2017 dataset (*without* SMOTE).

Model	Accuracy	Precision	Recall	F1-score	FAR	AUC
Naïve Bayes	0.8565	0.6364	0.9742	0.7699	0.1820	0.9711
Logistic Regression (LR)	0.9878	0.9722	0.9783	0.9752	0.0091	0.9961
LDA	0.9784	0.9426	0.9715	0.9569	0.0193	0.9923
Autoencoder (AE) + LR	0.9859	0.9691	0.9738	0.9715	0.0101	0.9960

Table 7. Performance metrics of all models on the CICIDS2017 dataset (*with* SMOTE on training data only).

Model	Accuracy	Precision	Recall	F1-score	FAR	AUC
Naïve Bayes	0.8564	0.6362	0.9742	0.7697	0.1821	0.9686
Logistic Regression (LR)	0.9862	0.9623	0.9824	0.9723	0.0126	0.9962
LDA	0.9735	0.9156	0.9830	0.9481	0.0296	0.9924
Autoencoder (AE) + LR	0.9806	0.9428	0.9805	0.9613	0.0194	0.9954

Key observations.

- *LR and AE+LR*: Both achieve a near-ceiling AUC (≈0.996). LR attains the best overall Accuracy/F1 (0.9878/0.9752) without SMOTE; AE+LR is a close second (0.9859/0.9715).
- *LDA*: Strong but consistently below LR/AE+LR in this dataset, with slightly higher FAR.
- *NB*: High Recall (≈0.97) but much lower Precision and a high FAR (≈0.18), yielding the lowest F1.

 doi:10.20944/preprints202511.0425.v1

- *Effect of SMOTE*: On CICIDS2017 (already balanced at the aggregate level after unification), SMOTE slightly *reduces* Accuracy/F1 for LR and AE+LR and increases FAR, which is consistent with minor overcompensation when the base split is relatively well balanced.

4.3. Takeaways

Across both datasets, NB provides a fast but conservative baseline with high Precision and low Recall. LDA is a strong classical competitor and slightly edges LR on NSL-KDD, whereas LR leads on CICIDS2017. The hybrid AE+LR consistently offers the best AUC and competitive F1, suggesting that unsupervised embeddings help linear classifiers capture non-linear structure without sacrificing interpretability. SMOTE yields modest gains in NSL-KDD (where minority attacks are scarce) and does not help—and can slightly hurt—on the unified CICIDS2017 split. These patterns align with prior IDS evidence: classical linear/probabilistic baselines remain valuable, and hybrid deep representations can further enhance linear decision functions under tabular traffic features.

4.3.1. Comparison with Related Work

It is instructive to situate our findings alongside recent IDS evaluations on NSL-KDD and CICIDS2017. Broadly, prior works report that deep learning (DL) models frequently attain near-ceiling discrimination on these corpora, whereas well-regularized linear/probabilistic baselines remain competitive, transparent references. A compact side-by-side comparison with recent IDS studies is summarized in Table 8.

Linear/Probabilistic baselines. Ali et al. [1] report LR around ∼97% accuracy and NB near ∼64% on their benchmark, trends that mirror our binary CICIDS2017 results where LR achieves 98.78% accuracy (F1 = 0.9752) and NB substantially trails due to its low precision and high FAR (Table 6). Our LDA is also competitive, consistent with surveys that position LR/LDA as strong, reproducible tabular baselines [26,27].

Deep learning and hybrids. Several studies show DL advantages on NSL-KDD/CICIDS2017. Umer and Abbasi [19] report an AE–LSTM two-stage system with ∼89% accuracy on NSL-KDD, prioritizing low false alarms; Chen et al. [20] obtain 86.8% on CICIDS2017 with a DBN+LSTM; and Roy et al. [21] reach 99.11% on CICIDS2017 using a lightweight supervised ensemble. Rather than training a full end-to-end DL classifier, our study adopts an *hybrid* approach (AE+LR): an autoencoder provides unsupervised embeddings and a linear LR supplies an auditable decision layer. This yields near-ceiling AUC on CICIDS2017 (AUC $\approx$ 0.996) and the top AUC on NSL-KDD among our four families (AUC $\approx$ 0.904), aligning with evidence that AE-driven representation learning can improve linear separability while preserving interpretability [6,18,26,27,29].

On metric deltas across papers. Absolute numbers in cross-paper comparisons routinely vary with dataset curation (binary vs. multi-class), train/test protocol (official splits vs. random), feature engineering and normalization, imbalance remedies, and leakage controls. For instance, our pipelines fit all preprocessing (and SMOTE when used) strictly on training data and evaluate on held-out splits, which can produce more conservative but comparable scores. Prior work also documents that SMOTE often helps minority recall on NSL-KDD but has mixed value on CICIDS2017 depending on the split [27,30]. Within this context, our results corroborate the prevailing ranking—NB $\ll$ LR/LDA $\lesssim$ DL—while showing that a simple AE+LR hybrid can deliver DL-like AUC with a transparent classifier.

 doi:10.20944/preprints202511.0425.v1

Table 8. Compact comparison with related IDS studies on NSL-KDD/CICIDS2017 (binary settings when reported).

Study	Dataset	Method	Headline metric (decimal, 4 d.p.)
Ali et al. [1]	Mixed	LR / DL (MLP/CNN)	LR Acc ≈ 0.9700; DL Acc ≈ 0.9800
Umer & Abbasi [19]	NSL-KDD	AE+LSTM	Acc ≈ 0.8900 (low FAR priority)
Chen et al. [20]	CICIDS2017	DBN+LSTM	Acc = 0.8680
Roy et al. [21]	CICIDS2017	Lightweight ensemble	Acc = 0.9911
This work	NSL-KDD	**AE+LR**	**AUC** = 0.9043**; F1** = 0.7583 **(with SMOTE; train-only)**
This work	CICIDS2017	**LR / AE+LR**	**Acc** = 0.9878 / 0.9859**; AUC** = 0.9961 / 0.9960 **(no SMOTE; train-only transforms)**

4.3.2. Key Takeaways

In summary, no single model is universally best for all deployment constraints. Among classical baselines, LR and LDA provide strong, transparent performance with low computational cost, making them attractive when interpretability and latency are priorities [27]. NB is generally not competitive on modern IDS corpora unless paired with additional design choices (e.g., feature engineering or hybridization; see Gu & Lu [14]).

Our hybrid pipeline (AE+LR) shows that using an autoencoder purely as an *unsupervised representation learner* can deliver near–state-of-the-art AUC on CICIDS2017 while preserving an auditable linear decision layer, aligning with reports that deep representations can improve linear separability without sacrificing explainability [26,27].

Operationally, the results underscore two practical points: (i) addressing class imbalance (e.g., with SMOTE applied *only* on training data) improves minority-class recall on NSL-KDD, whereas its benefit on CICIDS2017 depends on the split; and (ii) reporting FAR alongside Recall is essential to balance missed attacks against analyst workload. Overall, a layered IDS that uses fast statistical models for coarse filtering and learned representations for refined decisions can leverage complementary strengths under realistic constraints.

5. Conclusions and Future Work

This study presented a rigorous, reproducible comparison of three classical classifiers—Naïve Bayes (NB), Logistic Regression (LR), and Linear Discriminant Analysis (LDA)—and a hybrid *deep-representation* pipeline based on Autoencoder embeddings with a linear classifier (AE+LR). Using the NSL-KDD and CICIDS2017 benchmarks, we reported Accuracy, Precision, Recall, F1, AUC, and False Alarm Rate (FAR), with all preprocessing and any class rebalancing (SMOTE) fit strictly on the training split.

5.1. What Performed Best—and Where

On **NSL-KDD**, **AE+LR** and **LDA** offered the strongest overall trade-offs. AE+LR attained the highest *AUC* (≈0.904; Table 5) and the best F1 among the four models in both regimes (e.g., 0.7583 with SMOTE), while LDA slightly edged LR on Accuracy/F1 and AUC (e.g., F1 0.7537 vs. 0.7324 with SMOTE). **NB** achieved very high *Precision* (≈0.98) but very low *Recall* (≈0.24), yielding the weakest F1; its low FAR reflects a conservative decision boundary that misses many attacks.

On **CICIDS2017**, performance was strong for **LR**, **LDA**, and **AE+LR**. **LR** achieved the best *Accuracy/F1* without SMOTE (0.9878/0.9752), with **AE+LR** a close second (0.9859/0.9715), and both near-ceiling *AUC* (≈0.996). **NB** again trailed due to a much higher FAR (≈0.18) and lower Precision, despite high Recall.

5.2. Effect of Class Rebalancing

Applying **SMOTE** to training data produced *modest gains* on NSL-KDD (where rare attacks are genuinely scarce), e.g., small F1 improvements for LDA and AE+LR with essentially unchanged FAR.

 doi:10.20944/preprints202511.0425.v1

In the unified binary split of CICIDS2017, SMOTE did not help and in some cases slightly reduced Accuracy/F1 while increasing FAR—consistent with mild overcompensation when the base class balance is already adequate.

5.3. Practical Implications

For operators seeking *transparent* and *lightweight* deployment, **LDA** and **LR** remain strong, reproducible baselines. Where a small computational overhead is acceptable, **AE+LR** provides additional separability (highest AUC) and competitive F1 by leveraging an unsupervised representation while keeping a simple, auditable linear classifier. **NB** remains useful as a sanity-check baseline and for resource-constrained scenarios, but its low Recall on NSL-KDD cautions against relying on it as a stand-alone detector.

5.4. Limitations

Our primary focus was the binary setting, then, a complete multiclass analysis and cost-sensitive operating points were not the central objective here. In addition, AE was used strictly as a *feature learner* (not as a thresholded anomaly detector), so conclusions about reconstruction-threshold tuning are beyond scope.

5.4.1. Future Work

- **Multiclass and cost-aware evaluation.** Extend to full multiclass NSL-KDD with macro/micro metrics and per-class PR curves; explore cost-sensitive thresholds (e.g., penalizing FN for critical attacks).
- **Non-linear classical baselines.** Include Support Vector Machines (with kernels) and tree ensembles (Random Forest, Gradient Boosting) as non-linear yet interpretable tabular baselines.
- **Representation learning variants.** Compare AE with denoising/variational variants and different bottleneck sizes; quantify when AE+LR's gains over LR/LDA justify the added cost.
- **Imbalance strategies.** Evaluate cost-sensitive learning, probability calibration, and focal-type objectives; compare SMOTE with alternatives (SMOTE-NC, ADASYN) and informed undersampling.
- **Cross-dataset generalization.** Train on one domain and test on another (e.g., NSL-KDD $\rightarrow$ CICIDS2017 or UNSW-NB15) with domain adaptation/transfer learning; examine robustness to concept drift.
- **Operationalization and explainability.** Integrate local explanations (e.g., SHAP for LR/LDA and in the AE latent space) into alerting workflows; measure analyst triage time and trust.
- **Reproducible pipelines.** Release notebooks that regenerate all tables and reported metrics from `predict_proba/decision_function`; add regression tests to guard against library drift.
- **Cross-dataset validation.** We will replicate the full pipeline on UNSW-NB15 and CIC-IDS-2018 and run train$\rightarrow$test cross-domain transfers to assess robustness under dataset shift.
- **Dataset fusion and harmonization.** Explore concatenating harmonized feature-spaces from NSL-KDD and CICIDS2017 (with a domain indicator) to train unified models, and quantify trade-offs in accuracy/FAR vs. domain shift. Evaluate whether AE-based representations ease cross-domain fusion.

Overall, the results support a pragmatic recommendation: start with **LR/LDA** for simplicity and transparency, and adopt **AE+LR** when consistent separability gains (AUC) are desired without abandoning a linear, interpretable classifier. Careful use of imbalance remedies and alarm-centric metrics (Recall and FAR) is essential for reliable IDS deployment.

Author Contributions: Conceptualization, methodology, validation, investigation and data curation, Miguel Arcos-Argudo and Rodolfo Bojorque; writing—original draft preparation, Andrés Torres. All authors have read and agreed to the published version of the manuscript.

 doi:10.20944/preprints202511.0425.v1

Funding: This research was funded and supported by Universidad Politécnica Salesiana.

Data Availability Statement: The datasets analyzed in this study are publicly available and can be accessed through the following repositories: (1) The NSL-KDD Intrusion Detection Dataset, generated by the Canadian Institute for Cybersecurity at the University of New Brunswick [11], is available at: https://github.com/Jehuty4949/NSL_KDD (accessed on February 11, 2025). (2) The CICIDS2017 generated by the Canadian Institute for Cybersecurity at the University of New Brunswick [12]. An improved version is available at [25]: https://intrusion-detection.distrinet-research.be/CNS2022/CICIDS2017.html (accessed on February 5, 2025). A replication repository containing all source code and configuration files has been prepared and released in https://github.com/miguelarcosa/IDS-KDD-CICIDS2017.git.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. M. L. Ali, K. Thakur, S. Schmeelk, J. Debello, and D. Dragos, "Deep Learning vs. Machine Learning for Intrusion Detection in Computer Networks: A Comparative Study," *Applied Sciences*, vol. 15, no. 4, Art. no. 1903, Feb. 2025.
2. D. K. Roy and H. K. Kalita, "Enhanced deep autoencoder-based reinforcement learning model with improved flamingo search policy selection for attack classification," *Journal of Cybersecurity and Privacy*, vol. 5, no. 1, article 3, 2025. [Online]. Available: https://www.mdpi.com/2624-800X/5/1/3, doi: 10.3390/jcp5010003
3. Y. Song, S. Hyun, and Y.-G. Cheong, "Analysis of Autoencoders for Network Intrusion Detection," *Sensors*, vol. 21, no. 13, Art. no. 4294, 2021.
4. M. Elhamahmy, M. Ibrahim, M. Elsabagh, and H. Keshk, "Improving intrusion detection using LSTM-RNN to protect drones' networks," *Egyptian Informatics Journal*, vol. 27, no. 4, pp. 637–648, 2024.
5. T. Tayeh, S. Aburakhia, R. Myers, and A. Shami, "An Attention-Based ConvLSTM Autoencoder with Dynamic Thresholding for Unsupervised Anomaly Detection in Multivariate Time Series," *Machine Learning and Knowledge Extraction*, vol. 4, no. 2, pp. 350–370, 2022. [Online]. Available: https://doi.org/10.3390/make4020015
6. B. Lewandowski and R. Paffenroth, "Autoencoder Feature Residuals for Network Intrusion Detection: One-Class Pretraining for Improved Performance," *Machine Learning and Knowledge Extraction*, vol. 5, no. 3, pp. 868–890, 2023. [Online]. Available: https://doi.org/10.3390/make5030046
7. K. Kandi and A. García-Dopico, "Enhancing Performance of Credit Card Model by Utilizing LSTM Networks and XGBoost Algorithms," *Machine Learning and Knowledge Extraction*, vol. 7, no. 1, pp. 20–40, 2025. [Online]. Available: https://doi.org/10.3390/make7010020
8. N. Dash, S. Chakravarty, A. K. Rath, N. C. Giri, and N. Gowtham, "An optimized LSTM-based deep learning model for anomaly network intrusion detection," *Scientific Reports*, vol. 12, 2025 (in press).
9. M. Bacevicius, A. Paulauskaite-Taraseviciene, G. Zokaityte, L. Kersys, and A. Moleikaityte, "Comparative Analysis of Perturbation Techniques in LIME for Intrusion Detection Enhancement," *Machine Learning and Knowledge Extraction*, vol. 7, no. 1, pp. 21–41, 2025. [Online]. Available: https://www.mdpi.com/2504-4990/7/1/21
10. Q. Abbas, S. Hina, H. Sajjad, K. S. Zaidi, and R. Akbar, "Optimization of predictive performance of intrusion detection system using hybrid ensemble model for secure systems," *PeerJ Computer Science*, vol. 9, Art. e1552, 2023.
11. Canadian Institute for Cybersecurity (CIC), University of New Brunswick, "NSL-KDD Intrusion Detection Dataset." [Online]. Available: https://github.com/Jehuty4949/NSL_KDD (accessed Feb. 11, 2025).
12. Canadian Institute for Cybersecurity, "CICIDS2017 Dataset," University of New Brunswick, 2017. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html [Accessed: Feb. 5, 2025].
13. J. Fuhr, F. Wang, and Y. Tang, "MOCA: A network intrusion monitoring and classification system," *Journal of Cybersecurity and Privacy*, vol. 2, no. 3, pp. 629–639, 2022. [Online]. Available: https://www.mdpi.com/2624-800X/2/3/32, doi: 10.3390/jcp2030032
14. J. Gu and S. Lu, "An effective intrusion detection approach using SVM with naïve Bayes feature embedding," *Computers & Security*, vol. 103, Art. 102158, 2021.
15. B. Yasotha, T. Sasikala, and M. Krishnamurthy, "Wrapper Based Linear Discriminant Analysis (LDA) for Intrusion Detection in IIoT," *Computer Systems Science and Engineering*, vol. 45, no. 2, pp. 1625–1640, 2023.

16. H. Ghani, B. Virdee, and S. Salekzamankhani, "A deep learning approach for network intrusion detection using a small features vector," *Journal of Cybersecurity and Privacy*, vol. 3, no. 3, pp. 451–463, 2023. [Online]. Available: https://www.mdpi.com/2624-800X/3/3/23, doi: 10.3390/jcp3030023
17. J. Calle, R. Bojorque, A. Plaza, and P. Morquecho, "Detection of DDoS Attacks in Computer Networks Using Deep Learning," *Communications in Computer and Information Science*, vol. 2392 CCIS, pp. 291–305, 2026. doi: 10.1007/978-3-031-98287-3_21.
18. W. Xu, J. Jang-Jaccard, A. Singh, Y. Wei, and F. Sabrina, "Improving performance of autoencoder-based network anomaly detection on NSL-KDD dataset," *IEEE Access*, vol. 9, pp. 140136–140146, 2021.
19. Z. Umer and A. A. Abbasi, "A two-stage intrusion detection system with auto-encoder and LSTMs," *Applied Soft Computing*, vol. 121, Art. 108768, 2022.
20. A. Chen, Y. Fu, and X. Zheng, "An efficient network behavior anomaly detection using a hybrid DBN-LSTM network," *Computers & Security*, vol. 114, Art. 102600, 2022.
21. S. Roy, J. Li, B.-J. Choi, and Y. Bai, "A lightweight supervised intrusion detection mechanism for IoT networks," *Future Generation Computer Systems*, vol. 127, pp. 276–285, 2022.
22. C. A. de Souza, C. B. Westphall, and R. B. Machado, "Two-step ensemble approach for intrusion detection and identification in IoT and fog computing environments," *Computers & Electrical Engineering*, vol. 98, Art. 107694, 2022.
23. N. Gupta, V. Jindal, and P. B. Bedi, "CSE-IDS: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems," *Computers & Security*, vol. 112, Art. 102499, 2022.
24. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP)*, Madeira, Portugal, Jan. 2018, pp. 108–116.
25. L. Liu, G. Engelen, T. Lynar, D. Essam, and W. Joosen, "Error Prevalence in NIDS Datasets: A Case Study on CIC-IDS-2017 and CSE-CIC-IDS-2018," in *Proc. IEEE Conf. on Communications and Network Security (CNS)*, 2022, pp. 254–262.
26. F. Yan, S. Wen, S. Nepal, C. Paris, and Y. Xiang, "Explainable Machine Learning in Cybersecurity: A Survey," *International Journal of Intelligent Systems*, vol. 37, no. 12, pp. 12305–12334, 2022.
27. Y. Li, L. Wang, and S. Zhang, "A Comprehensive Survey on Intrusion Detection Algorithms," *Computers & Electrical Engineering*, vol. 121, Art. no. 109863, 2025.
28. R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection," *Electronics*, vol. 8, no. 3, Art. no. 322, 2019.
29. A. T. Azar, E. Shehab, A. M. Mattar, I. A. Hameed, and S. A. Elsaid, "Deep Learning Based Hybrid Intrusion Detection Systems to Protect Satellite Networks," *Journal of Network and Systems Management*, vol. 31, no. 4, Art. no. 82, 2023.
30. K. Sayegh, S. Khanduja, and A. G. J. Holt, "Enhanced Intrusion Detection with LSTM-Based Model, Feature Selection, and SMOTE for Imbalanced Data," *Applied Sciences*, vol. 14, no. 2, Art. no. 479, 2024.
31. Scikit-learn Developers, "`LinearDiscriminantAnalysis`—User Guide and API," *scikit-learn Documentation*, release 1.7 (and prior), accessed 2025.