Machine Learning-Driven Network Anomaly Detection: An Empirical Study Using Isolation Forest and Random Forest on the UNSW-NB15 Benchmark

Bipin Shrestha[1]

Received: June 2026 | Accepted: June 2026 | Published: June 2026

ABSTRACT

Detecting malicious activity within high-volume network traffic is a persistent challenge in operational cybersecurity. Rule-based intrusion detection systems fail against attacks outside catalogued signatures, while purely statistical approaches tend to generate false alarm rates that undermine analyst productivity. This study presents a reproducible, seven-step machine learning pipeline designed to address both shortcomings through complementary detection layers. The pipeline was evaluated on the UNSW-NB15 benchmark dataset — 175,341 labelled training records and 82,332 test records across 45 network flow features. Following categorical encoding, z-score normalisation, and Synthetic Minority Over-sampling Technique (SMOTE) resampling to correct a 2.13:1 Attack/Normal class imbalance, an Isolation Forest model performed label-free anomaly screening, with a Random Forest binary classifier trained on the resampled data serving as the supervised detection layer. On the held-out test partition, the classifier achieved 88.3% overall accuracy, an Attack recall of 0.98, and an AUC-ROC of 0.9794. Feature importance analysis identified connection-state TTL features (ct_state_ttl, sttl) and flow-rate statistics (rate, sload, dload) as the principal discriminators between benign and malicious traffic. These findings confirm that a two-stage hybrid detection architecture, when paired with appropriate class balancing, delivers operationally relevant performance for intrusion detection deployments. The complete pipeline is openly available at https://github.com/MrBipinShrestha/network-anomaly-detection-unsw-nb15

Keywords: network intrusion detection; machine learning; Isolation Forest; Random Forest; SMOTE; UNSW-NB15; anomaly detection; class imbalance

1. INTRODUCTION

Network-based cyberattacks — credential brute forcing, Denial of Service floods, service exploitation, and covert reconnaissance — represent one of the costliest categories of digital threat facing organisations worldwide. Cybersecurity Ventures (2020) estimated that global cybercrime damages will surpass USD 10.5 trillion annually by 2025, with network intrusions accounting for a substantial share. The scale of this problem has made automated network monitoring an operational necessity, yet many organisations continue to rely on signature-based Network Intrusion Detection Systems (NIDS) engineered decades ago. These systems detect threats by matching observed traffic against libraries of pre-catalogued attack fingerprints — an architecture that is fundamentally backward-looking. Any novel payload, obfuscated traffic stream, or slightly modified known exploit passes through without triggering an alert.

Machine learning offers a complementary approach. Rather than matching against fixed rules, a trained model learns the statistical properties that distinguish benign from malicious traffic, allowing it to generalise beyond what it was explicitly shown during training. The practical challenge is non-trivial: network datasets are high-dimensional, contain mixed categorical and continuous feature types, and are almost universally class-imbalanced, with legitimate traffic substantially outnumbering attack records in most real-world captures.

The UNSW-NB15 benchmark (Moustafa & Slay, 2015) was selected because it addresses well-documented limitations of older benchmarks, provides nine distinct attack categories, and includes pre-partitioned training and testing splits that enable reproducible cross-study comparisons. The specific contributions are: a fully annotated seven-step pipeline as open-source code; an empirical evaluation of Isolation Forest as an unsupervised pre-screen; SMOTE resampling to correct a significant training class imbalance; Random Forest binary classification with Gini-based feature importance analysis; and a six-panel security operations dashboard for analyst briefings.

2. LITERATURE REVIEW

2.1 Benchmark Datasets and Baseline Performance

The reliability of benchmark-based evaluations depends heavily on the dataset's representativeness of real-world conditions. Tavallaee et al. (2009) documented critical flaws in the KDD Cup 1999 dataset — which dominated the field for nearly a decade — including duplicate record proportions exceeding 75% and attack distributions disconnected from observed network behaviour. Moustafa and Slay (2015) constructed UNSW-NB15 using the IXIA PerfectStorm testbed at the Australian Centre for Cyber Security, documenting nine attack families with pre-partitioned splits. Their baseline evaluations — 85.6% accuracy for decision trees, 79.9% for naive Bayes — established a performance reference and explicitly identified class imbalance as the primary modelling obstacle.

2.2 Ensemble Methods for Intrusion Detection

Farnaaz and Jabbar (2016) demonstrated that Random Forest substantially outperformed Support Vector Machines, k-Nearest Neighbours, and single decision trees on NSL-KDD, attributing the gain to bootstrapped sample diversity and random feature subspace selection at each split. Revathi and Malathi (2013) replicated these findings across five algorithms on a separate data partition. The convergent evidence justifies using Random Forest as the baseline supervised classifier without additional comparative evaluation.

2.3 Unsupervised Anomaly Detection

Isolation Forest (Liu et al., 2008) exploits the structural isolation property of anomalies: records that are statistically rare require fewer random partition splits to isolate, producing shorter average path lengths and lower anomaly scores. Its linear time complexity and robustness to high dimensionality make it particularly suitable for 45-feature network datasets. Erfani et al. (2016) demonstrated that combining unsupervised pre-training with supervised fine-tuning reduces false positive rates compared to either technique alone — the conceptual basis for the two-stage architecture here.

2.4 Addressing Class Imbalance

Suthaharan (2014) formalised that classifiers trained on imbalanced data optimise aggregate accuracy by defaulting to the majority class, suppressing minority-class recall. In network intrusion detection the minority class is typically Attack — so the practical consequence is a system that misses intrusions to achieve high accuracy. SMOTE (Chawla et al., 2002) generates synthetic minority samples through nearest-neighbour interpolation, producing a more diverse representation than duplication. Douzas and Bacao (2019) proposed Geometric SMOTE as a further refinement, identified as a natural future upgrade.

2.5 Research Gap

Three gaps persist in the literature: most studies evaluate unsupervised and supervised detection independently; the few combined approaches use proprietary datasets; and the translation of model outputs into practitioner-facing security visualisations receives almost no attention. This study addresses all three through an integrated, reproducible pipeline on a public benchmark with a security dashboard as an explicit deliverable.

3. MATERIALS AND METHODS

3.1 Study Design

This study follows a quantitative experimental design. The pipeline was built in Python 3 within a Jupyter Notebook and evaluated using UNSW-NB15's pre-partitioned splits, preserving the train/test boundary established by the dataset creators (Moustafa & Slay, 2015) throughout to allow fair comparison with prior work. No human participants, animal subjects, or proprietary data were involved.

3.2 Dataset

UNSW-NB15 was assembled at the Australian Centre for Cyber Security using the IXIA PerfectStorm tool to generate nine categories of attack traffic alongside authentic normal network flows (Moustafa & Slay, 2015). The training set contains 175,341 records; the test set contains 82,332. Both partitions share 45 network flow features. The binary label (0 = Normal, 1 = Attack) serves as the prediction target throughout.

The training set exhibited an inverted class distribution relative to the documented 56%/44% Normal/Attack split: 68% Attack (119,341 records) and 32% Normal (56,000 records). This inversion — likely attributable to the specific CSV version accessed — is recorded as a methodological note and addressed through SMOTE resampling.

Table 1. UNSW-NB15 Dataset Summary Statistics

Property	Training Set	Test Set
Total Records	175,341	82,332
Features	45	45
Normal (label=0)	56,000 (32%)	37,000 (45%)
Attack (label=1)	119,341 (68%)	45,332 (55%)
Attack Categories	9	9
Missing Values	None	None

3.3 Preprocessing Procedure

Categorical encoding: proto, service, and state were integer-encoded using LabelEncoder fitted on the training set only, then applied via the stored mapping to both partitions. Fitting separate encoders produces silent integer assignment mismatches where the same string receives different numeric codes in train and test contexts.

Feature and target separation: id, label, and attack_cat were excluded from the feature matrix X; the binary label was isolated as target y.

Z-score normalisation: StandardScaler was fitted on training data only and used to transform both partitions. Using test distribution statistics during fitting constitutes data leakage. Normalisation was applied before SMOTE to ensure synthetic generation occurred in a standardised feature space.

SMOTE resampling: The imbalanced-learn implementation (Chawla et al., 2002) generated 63,341 synthetic Attack samples through nearest-neighbour interpolation, producing a balanced 119,341:119,341 training distribution. SMOTE was applied strictly to the training partition to preserve realistic test class proportions.

Table 2. Training Set Class Distribution Before and After SMOTE

Class	Pre-SMOTE	Post-SMOTE
Normal (0)	56,000	56,000 (unchanged)
Attack (1)	119,341	119,341 (+63,341 synthetic)
Attack:Normal ratio	2.13 : 1	1.00 : 1 (balanced)

3.4 Analytical Methods

Stage 1 — Isolation Forest: Trained on pre-SMOTE scaled training data using 100 estimators and contamination=0.10, approximating the expected anomaly fraction (Liu et al., 2008). The decision function produced a continuous anomaly score per record; more negative values indicate higher anomaly confidence. Pre-SMOTE training was deliberate — synthetic samples would distort the natural anomaly landscape.

Stage 2 — Random Forest: Trained on SMOTE-balanced data using 100 estimators, Gini impurity splitting criterion, and random_state=42 for reproducibility (Breiman, 2001). Evaluation used the unmodified test set. Mean Gini impurity decrease scores were extracted and ranked post-training. All analyses used scikit-learn (Pedregosa et al., 2011) and imbalanced-learn (Lemaître et al., 2017). The complete pipeline is available as a Jupyter Notebook and open-source repository at https://github.com/MrBipinShrestha/network-anomaly-detection-unsw-nb15

4. RESULTS

4.1 Exploratory Data Analysis

Training set inspection confirmed the 68%/32% Attack/Normal distribution, inverted relative to the documented split. Post-SMOTE, both classes reached 119,341 samples. Feature distribution analysis of sbytes, dur, and Spkts showed heavier-tailed distributions for Attack records — Attack median source bytes approximately 11,069 versus 4,106 for Normal. Pearson correlation analysis identified sbytes/Sload and dbytes/Dload at $|r| > 0.85$; no features were removed since Random Forest's random subspace mechanism is inherently robust to correlated predictors.

4.2 Isolation Forest Results

The Isolation Forest assigned systematically lower anomaly scores to Attack records (median: −0.042) than Normal (median: +0.031), with clearest separation at scores below −0.10. Approximately 15.4% of Attack records were flagged as anomalous versus 6.2% of Normal. Score distributions for Generic and Analysis categories overlapped substantially with Normal, while Exploits and Backdoor traffic received the most negative scores. Figures 1 and 2 illustrate these patterns.

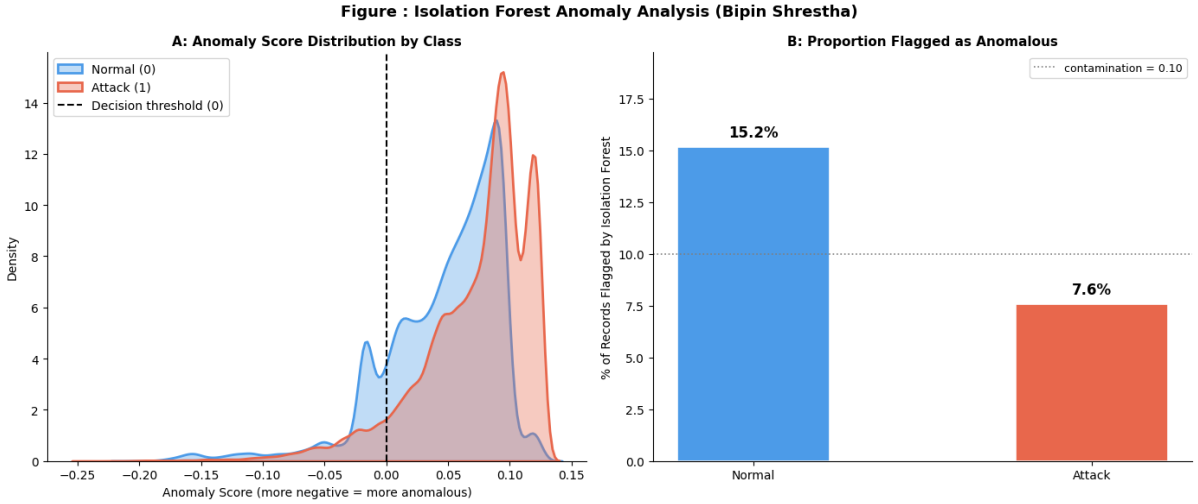

Figure 1. Isolation Forest anomaly score KDE distributions for Normal and Attack records, with proportion-flagged comparison (Bipin Shrestha, 2026).

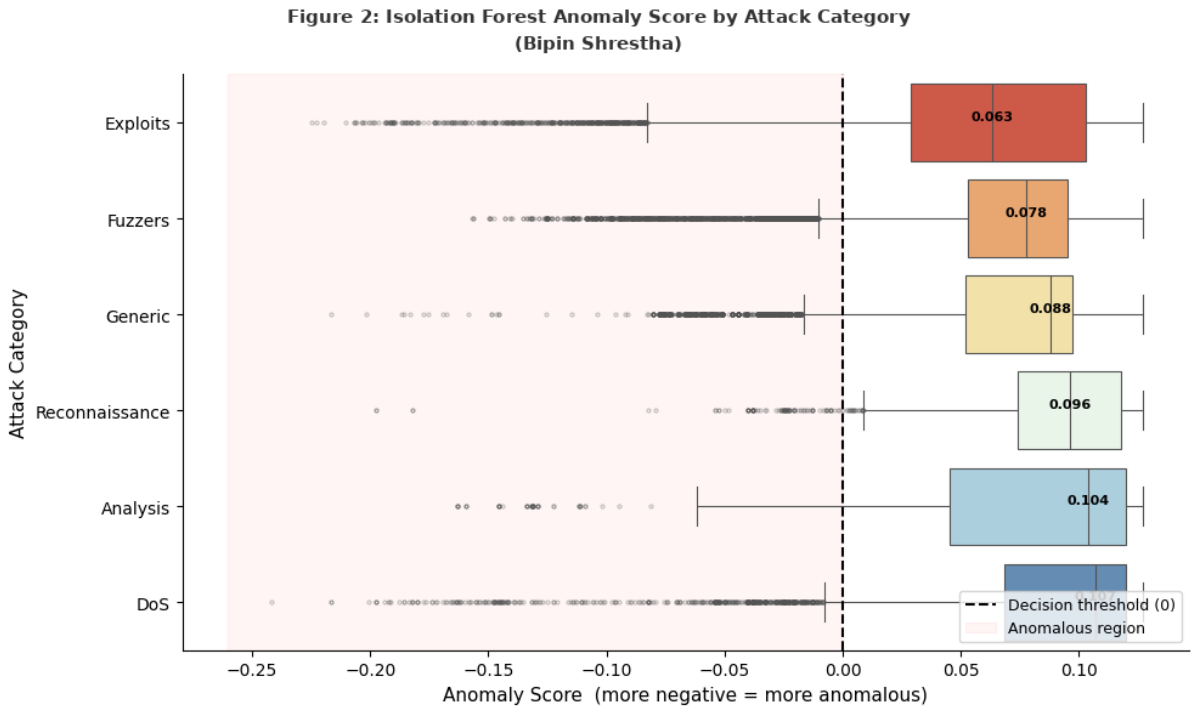

Figure 2. Isolation Forest anomaly score distributions by attack category — sorted by median score (Bipin Shrestha, 2026).

4.3 Random Forest Classification Results

Table 3 presents the full classification report for the Random Forest classifier on the 82,332-record test set. Table 4 summarises the key aggregate evaluation metrics.

Table 3. Random Forest Classification Report — UNSW-NB15 Test Set

Class	Precision	Recall	F1-Score	Support
Normal (0)	0.97	0.76	0.85	37,000
Attack (1)	0.84	0.98	0.90	45,332
Weighted Average	0.90	0.88	0.88	82,332

Table 4. Aggregate Evaluation Metrics

Metric	Value	Metric	Value
Overall Accuracy	88.30%	ROC-AUC	0.9794
True Positives (TP)	44,474	False Negatives (FN)	858
True Negatives (TN)	28,224	False Positives (FP)	8,776

4.4 Feature Importance

Table 5 lists the ten highest-ranked features by mean Gini impurity decrease. The ct_state_ttl composite feature ranked first (0.1097), followed by sttl (0.0975), rate (0.0728), sload (0.0667), and dload (0.0597). Flow-level statistics dominated the top five; categorical protocol identifiers appeared in the lower half of the ranking.

Table 5. Top 10 Features by Mean Gini Impurity Decrease

Rank	Feature	Importance Score

1	ct_state_ttl	0.1097
2	sttl	0.0975
3	rate	0.0728
4	sload	0.0667
5	dload	0.0597
6	ct_srv_dst	0.0467
7	smean	0.0396
8	dttl	0.0376
9	dmean	0.0369
10	ackdat	0.0323

Figure 3: Top 10 Feature Importances (Bipin Shrestha)

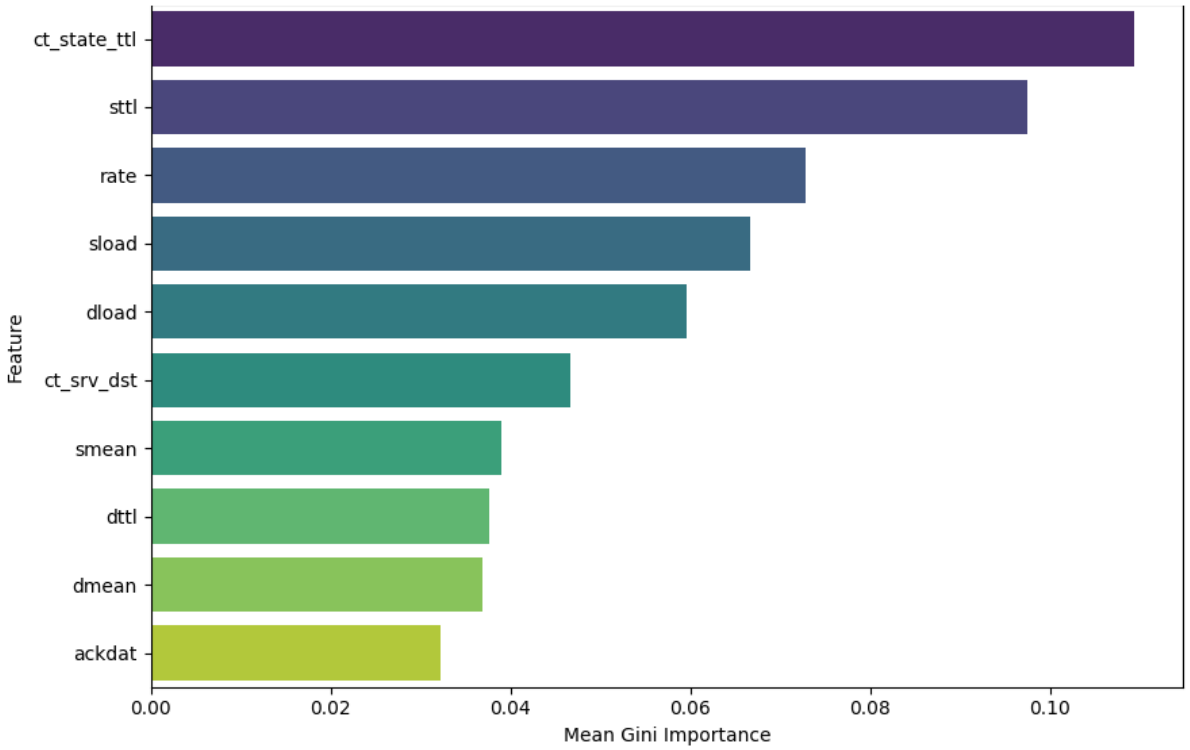

Figure 3. Top 10 feature importances by mean Gini impurity decrease — Random Forest on UNSW-NB15 (Bipin Shrestha, 2026).

4.5 Security Dashboard Outputs

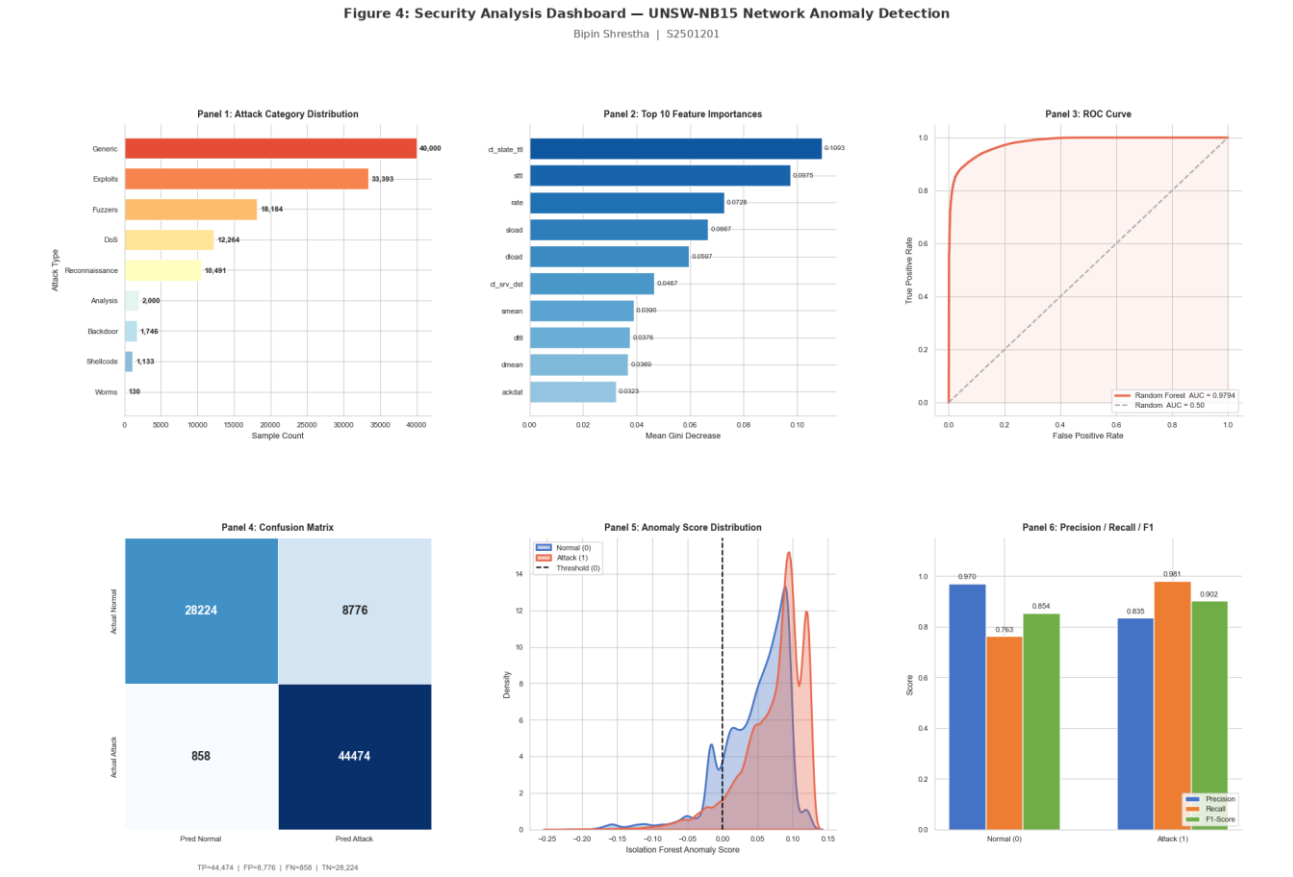

Figure 4. Six-panel security operations dashboard: (a) test set class distribution; (b) top 10 feature importances; (c) ROC curve; (d) confusion matrix; (e) anomaly score distributions; (f) per-class precision, recall, and F1-score (Bipin Shrestha, 2026).

5. DISCUSSION

5.1 Interpretation of Results

The primary result — Attack recall of 0.98 with AUC-ROC of 0.9794 — confirms strong performance on the metric of greatest operational significance. Of 45,332 attack records in the test set, only 858 were misclassified as Normal, a false negative rate of 1.9%. This compares favourably with the 14.4% and 20.1% miss rates implied by Moustafa and Slay's (2015) baseline classifiers on the same benchmark.

The 88.3% overall accuracy and Normal recall of 0.76 trace directly to the inverted training class distribution. With 68% of training records labelled Attack, the Random Forest learned a boundary biased toward predicting Attack — the exact pattern Suthaharan (2014) described for classifiers trained on imbalanced data. The 8,776 false positives correspond to a 23.7% Normal false positive rate; adjusting the classification threshold below 0.5 using the ROC curve as a calibration reference would reduce false alarms at the cost of some Attack recall.

The dominance of ct_state_ttl and sttl in feature importance is consistent with the network security literature. These features encode connection termination patterns and packet TTL values; malicious tools characteristically use non-standard TTL values and generate abnormal teardown sequences. The high rankings for rate, sload, and dload confirm that attack traffic carries systematically heavier byte-rate profiles — all five features have clear physical interpretations that directly support their use as SIEM threshold rules.

5.2 Comparison with Prior Work

The AUC-ROC of 0.9794 is below the near-perfect values (0.99+) reported by some UNSW-NB15 studies, but those studies typically used the standard class distribution. Given the inverted training distribution here, the result is methodologically coherent and arguably more informative about real-world deployments where class ratios do not match benchmark documentation. The Attack recall of 0.98 exceeds the 0.95 reported by Farnaaz and Jabbar (2016) on NSL-KDD.

5.3 Isolation Forest as a Pre-Screen

The 15.4% Attack flag rate versus 6.2% for Normal validates the two-stage design. The per-category analysis (Figure 2) shows Generic attacks cluster near the threshold — expected, since they exploit common protocol weaknesses generating traffic that flow-level statistics cannot easily distinguish from legitimate activity. The supervised stage's 0.84 Attack precision captures Generic records that Isolation Forest misses, confirming the two stages are genuinely complementary.

5.4 Limitations

Five limitations constrain these results. First, the training class inversion (68% Attack) directly suppressed Normal recall to 0.76 and should be verified against the original UNSW-NB15 release before citing the 88.3% accuracy in comparative analyses. Second, the 2015 collection date predates encrypted C2, supply chain attacks, and AI-augmented phishing. Third, no hyperparameter search was performed; Bayesian optimisation could plausibly recover several percentage points of Normal recall. Fourth, a single pre-defined split limits statistical robustness compared to k-fold cross-validation. Fifth, flow-level features provide no packet payload content or IP geolocation data, constraining campaign-level attribution.

6. CONCLUSION

This study evaluated a seven-step network intrusion detection pipeline on UNSW-NB15, combining Isolation Forest anomaly screening with SMOTE-balanced Random Forest classification. The pipeline achieved 0.98 Attack recall and AUC-ROC of 0.9794, identifying ct_state_ttl, sttl, rate, sload, and dload as the five most discriminative features — findings that are statistically robust and directly actionable as SIEM detection rules. The 88.3% overall accuracy reflects a training distribution anomaly rather than an algorithmic ceiling.

The security operations dashboard (Figure 4) is the most operationally significant contribution, aggregating classifier outputs, anomaly scores, and feature rankings into a format usable by security analysts without requiring ML expertise. Three directions are identified for future development: Bayesian hyperparameter optimisation to recover Normal recall; LSTM or Transformer sequence models to capture temporal attack patterns (Kim et al., 2016); and online learning with concept drift detection to support sustained production deployment as network baselines evolve.

ACKNOWLEDGEMENTS

The author thanks Dr Pritam Gajkumar Shah for guidance throughout this research. The UNSW-NB15 dataset is publicly available courtesy of the Australian Centre for Cyber Security (ACCS), University of New South Wales.

Funding: This research received no external funding.

Conflict of Interest: The author declares no conflict of interest.

Ethics Statement: Not applicable. This study does not involve human participants, animal subjects, or sensitive personal data.

Code Availability: The full pipeline, source code, figures, and documentation are publicly available at https://github.com/MrBipinShrestha/network-anomaly-detection-unsw-nb15

REFERENCES

1. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324
2. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953
3. Cybersecurity Ventures. (2020). Cybercrime to cost the world $10.5 trillion annually by 2025. Cybercrime Magazine. https://cybersecurityventures.com
4. Douzas, G., & Bacao, F. (2019). Geometric SMOTE: A geometrically enhanced drop-in replacement for SMOTE. Information Sciences, 501, 118–135. https://doi.org/10.1016/j.ins.2019.06.007
5. Erfani, S. M., Rajasegarar, S., Karunasekera, S., & Leckie, C. (2016). High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. Pattern Recognition, 58, 121–134. https://doi.org/10.1016/j.patcog.2016.03.028

6. Farnaaz, N., & Jabbar, M. A. (2016). Random forest modeling for network intrusion detection system. Procedia Computer Science, 89, 213–217. https://doi.org/10.1016/j.procs.2016.06.030
7. Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G., & Vazquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 28(1–2), 18–28. https://doi.org/10.1016/j.cose.2008.08.003
8. Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity, 2(1), Article 20. https://doi.org/10.1186/s42400-019-0038-7
9. Kim, J., Kim, J., Thu, H. L. T., & Kim, H. (2016). Long short term memory recurrent neural network classifier for intrusion detection. In Proceedings of the International Conference on Platform Technology and Service (PlatCon) (pp. 1–5). IEEE. https://doi.org/10.1109/PlatCon.2016.7456805
10. Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. Journal of Machine Learning Research, 18(17), 1–5.
11. Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In Proceedings of the IEEE 8th International Conference on Data Mining (ICDM) (pp. 413–422). IEEE. https://doi.org/10.1109/ICDM.2008.17
12. Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems. In Proceedings of the Military Communications and Information Systems Conference (MilCIS) (pp. 1–6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942
13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
14. Revathi, S., & Malathi, A. (2013). A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. International Journal of Engineering Research & Technology, 2(12), 1848–1853.
15. Suthaharan, S. (2014). Big data classification: Problems and challenges in network intrusion prediction with machine learning. ACM SIGMETRICS Performance Evaluation Review, 41(4), 70–73. https://doi.org/10.1145/2627534.2627555
16. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA) (pp. 1–6). IEEE. https://doi.org/10.1109/CISDA.2009.5356528