From Detection to Defence: Classification Modelling of Network Intrusion Attacks

Prepared by

Rebecca Stalley-Moores

From Detection to Defence: Classification Modelling of Network Intrusion Attacks.

Prepared by: **Rebecca Stalley-Moores**

Date: **11/02/2026**

1. Executive Summary

This analysis developed a multi-class classification model for network intrusion detection using the NSL-KDD cybersecurity dataset, evaluating the effectiveness of multiple machine learning approaches in identifying five traffic categories: Normal, DoS, Probe, R2L, and U2R attacks.

Key Findings: Exploratory analysis revealed extreme class imbalance as the dominant challenge, with Normal and DoS traffic comprising 90% of training data while the most dangerous attack types — R2L (remote-to-local) and U2R (user-to-root) — represented just 0.79% and 0.04% respectively. Feature analysis identified distinct attack signatures: connection state features (`flag_S0`) proved the strongest overall discriminator, while session-level features (`root_shell, logged_in`) were critical for detecting rare but high-severity intrusions. Notably, features that appeared near-constant in overall distribution proved to be among the most important for minority class detection, validating their retention during feature engineering.

Model Development: Through iterative refinement — including SMOTE oversampling, cost-sensitive learning with balanced and custom domain-motivated weights, and hyperparameter tuning via RandomizedSearchCV — XGBoost with balanced class weights emerged as the recommended model. The tuned model achieved a macro-averaged F1 of 0.641, macro-averaged recall of 0.618, and overall accuracy of 80.7%, outperforming all 16 other model-strategy combinations on four of six evaluation criteria. Cost-sensitive learning proved more effective than synthetic oversampling for extreme minority classes, with balanced class weights delivering larger performance gains than SMOTE from just 52 real U2R training samples.

Research Impact: The analysis demonstrates that handling class imbalance matters more than model selection in severely imbalanced domains. R2L recall improved from 6.9% to 29.6% and U2R recall from 3.5% to 18.0% through the combined effect of class reweighting and hyperparameter tuning — without any change to the underlying algorithm. Feature importance analysis confirmed that the model's decision-making aligns with established cybersecurity domain knowledge, with the top five features accounting for 45.7% of total model importance and mapping directly to known attack signatures.

Strategic Next Steps: Future work should focus on three priorities: incorporating unsupervised anomaly detection to complement the supervised classifier for novel attack types not represented in

training data, applying this methodology to modern intrusion detection datasets (such as CICIDS2017) that reflect contemporary network protocols and attack vectors, and exploring advanced resampling techniques such as ADASYN or borderline-SMOTE to improve synthetic sample quality for extreme minority classes.

2. Objectives

This analysis aims to: (1) develop and evaluate multiple classification models for network intrusion detection, comparing their performance across attack categories; (2) optimise models to minimise false negatives (missed attacks) while maintaining acceptable false positive rates, recognising that in cybersecurity applications undetected threats pose greater risk than false alarms; and (3) identify the most important predictive features to support model interpretation and provide actionable insights for network security stakeholders.

2.1. Primary Objectives:

1. **Conduct exploratory data analysis** to understand feature distributions, identify patterns distinguishing attack types, and detect data quality issues including class imbalance
2. **Build and compare multiple classification models** including Logistic Regression (baseline), Random Forest, and XGBoost to identify the best performing approach
3. **Evaluate model performance** with emphasis on recall and F1-score to minimise missed attacks, alongside precision, accuracy, and ROC-AUC
4. **Refine models through iterative improvements** including class imbalance handling and hyperparameter tuning to optimise predictive performance
5. **Identify key predictive features** for interpretation and actionable security insights.

2.2. Success Criteria:

- **High recall** across all attack categories, particularly for high-risk minority classes (U2R, R2L), ensuring critical attacks are not missed
- **Balanced performance** maintaining acceptable precision to limit false alarms
- **Effective class imbalance management** demonstrated through improved minority class F1-scores
- **Selection of final model** that balances predictive performance with interpretability
- **Reliable predictions** suitable for real-time intrusion detection systems

3. Data Summary

- The dataset is sourced from the NSL-KDD (Network Security Laboratory - Knowledge Discovery in Databases) dataset, available via Kaggle and the University of New Brunswick's Canadian Institute for Cybersecurity.
- The original dataset contains 125,973 training records and 22,544 test records, with 43 columns representing network connection features from simulated network traffic.
- Features are organized hierarchically to capture network behaviour at different aggregation levels.

        - individual connection metrics (e.g., duration, src_bytes)
        - service-level aggregations over 2-second windows (srv_ prefix)
        - host-level aggregations over 100 connections (dst_host_ prefix)
        - combined host-service aggregations (dst_host_srv_ prefix).
    - This hierarchy enables detection of different attack patterns: DoS attacks generate anomalies at all levels through widespread flooding, Probe attacks show host-level patterns from targeted scanning, while R2L and U2R attacks exhibit focused host-service patterns targeting specific vulnerabilities
    - In its original format there are 38 integer columns, 3 float columns (rate features), and 4 object columns (categorical features).

- A typical variable example:

    - `num_failed_logins`: Count of failed login attempts (0-5, integer)
    - `serror_rate`: Percentage of connections with SYN errors (0.0-1.0, float)
    - `protocol_type`: Network protocol used (tcp, udp, icmp, categorical)

- The dataset contains 23 specific attack types that were mapped to 5 main categories for classification:

    - **Normal**: Legitimate network traffic (67,343 records, 53.5%)
    - **DoS** (Denial of Service): Attacks overwhelming systems (45,927 records, 36.5%)
    - **Probe**: Surveillance and reconnaissance attacks (11,656 records, 9.3%)
    - **R2L** (Remote to Local): Unauthorized remote access (995 records, 0.8%)
    - **U2R** (User to Root): Privilege escalation attacks (52 records, 0.04%)

- The target variable for analysis is `attack_category`, representing the categorised type of network connection (Normal or attack type).

4. Data Cleaning and Feature Engineering

4.1. Data Cleaning

- **Missing values:** The NSL-KDD dataset contained no missing values. All 43 columns were complete for both training and test sets.
- **Duplicates**: No duplicate rows were found in either the training or test datasets.
- **Data types**: Data types were verified and found to be correct—41 numeric features (38 integers, 3 floats) and 5 categorical features (3 network attributes plus target and difficulty level).
- **Redundant columns:** Two columns were dropped as they were not genuine network features:
    - `attack_type`: Redundant after mapping to `attack_category` (5 classes)
    - `difficulty_level`: Dataset metadata (difficulty rating 0-21), not a real network feature

- **Near-constant features:** A systematic check identified 13 features where >99% of values were identical. These were evaluated for their ability to discriminate between attack types:

Feature	Most Common %	Variance	Decision	Rationale
num_outbound_cmds	100.00%	0.000000	Drop	Zero variance - completely useless
is_host_login	100.00%	0.000008	Drop	Only 1 non-zero record in entire dataset
land	99.98%	0.000198	Drop	Only 25 non-zero records (0.02%)
su_attempted	99.94%	0.002039	Drop	No discrimination between attack types (0.12% Normal vs 0.10% R2L)
num_shells	99.96%	0.000492	Keep	9.62% of U2R attacks have non-zero values - strong signal for privilege escalation
num_failed_logins	99.90%	0.002047	Keep	5.23% of R2L attacks - signals unauthorized access attempts
urgent	99.99%	0.000206	Keep	1.92% of U2R attacks - small but potentially useful signal

- **Counter limit censoring:** Network monitoring tools use fixed-size counters with maximum values. Analysis revealed that two features were severely affected by this limitation:

Feature	Counter Limit	Records at Cap	Percentage Censored
dst_host_count	255 (8-bit: 2^8-1)	74,099	58.82%
dst_host_srv_count	255 (8-bit: 2^8-1)	35,993	28.57%
count	511 (9-bit: 2^9-1)	1,437	1.14%
srv_count	511 (9-bit: 2^9-1)	1,012	0.80%

Decision: **All counter-limited features were retained despite censoring because:**

- Even capped values (255, 511) provide discriminative signal
- High connection counts are indicative of attacks (especially DoS)
- This is a standard limitation in network monitoring datasets
- The limitation affects data interpretation but not predictive utility
- **Logical consistency checks:** The following validation checks confirmed data integrity:
  - **Negative values**: No negative values found in count or size features (`duration`, `src_bytes`, `dst_bytes`, `count`, `srv_count`)
  - **Rate features**: All rate features confirmed to be within valid range [0, 1]
  - **Binary features**: All binary indicators (`logged_in`, `root_shell`, `is_guest_login`) contained only values 0 or 1

- **Final dataset shape:** After data cleaning:
  - Training set: 125,973 rows × 38 columns (37 features + 1 target)
  - Test set: 22,544 rows × 38 columns (37 features + 1 target)
  - **Features dropped**: 6 total (2 redundant, 4 near-constant)
  - **Features retained**: 37 features for modelling

- Data quality notes:
    - **Class imbalance**: Severe imbalance identified with minority classes U2R (52 records, 0.04%) and R2L (995 records, 0.8%). This will be addressed through SMOTE (Synthetic Minority Over-sampling Technique) during model refinement.
    - **Counter censoring**: dst_host_count and dst_host_srv_count are subject to significant censoring, with true values unknown for 58.82% and 28.57% of records respectively. This limitation is noted for report discussion but does not preclude feature usage.
    - **Dataset quality**: The NSL-KDD dataset is well-curated with no missing values, no duplicates, and no logical inconsistencies, making it suitable for classification modelling.

4.2. Data Wrangling and Feature Engineering

4.2.1. Attack Type Mapping

The original `attack_type` variable (23 specific attacks) was mapped to 5 high-level attack categories for classification:

Attack Category	Attack Types Included	Training Records	Percentage
Normal	normal	67,343	53.46%
DoS	back, land, neptune, pod, smurf, teardrop	45,927	36.46%
Probe	ipsweep, nmap, portsweep, satan	11,656	9.25%
R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster	995	0.79%
U2R	buffer_overflow, loadmodule, perl, rootkit	52	0.04%

These mappings group attacks by their primary objective and mechanism:

- DoS (Denial of Service) - All attacks that overwhelm system resources to deny legitimate access
- Probe - Reconnaissance attacks that scan for vulnerabilities or gather network information
- R2L (Remote to Local) - Exploits that gain unauthorized access from a remote location
- U2R (User to Root) - Privilege escalation attacks that elevate from normal user to administrator/root access

4.2.2. Categorical Encoding

Service Feature (70 categories)

Challenge: High cardinality categorical variable with extreme variance in attack rates (0% to 100%)

Analysis findings:

- Top 10 services cover 80.61% of data, remaining 60 services represent long tail
- Attack rates range from 0% (tftp_u, ntp_u, red_i) to 100% (mtp, nntp, klogin, kshell, ldap, etc.)

- Global attack rate: 46.54%
- 15 rare services with <100 records show distinct patterns (e.g., urh_i 100% benign vs pm_dump 100% attack)

Encoding approach: Smoothed target encoding

- Formula: `(n × category_mean + m × global_mean) / (n + m)`, where m=100 (smoothing parameter)
- Each service converted to its historical attack rate (0-1 continuous scale)
- Smoothing regularizes rare services toward global mean to prevent overconfidence

Effect of smoothing on example services:

Service	Records	Raw Attack Rate	Smoothed Rate	Difference
pm_dump	5	100.0%	49.1%	-50.9%
urh_i	10	0.0%	42.3%	+42.3%
http	40,338	5.7%	5.8%	+0.1%
private	21,853	95.5%	95.3%	-0.2%

Rationale: Target encoding reduces dimensionality (70 categories → 1 continuous feature) while capturing the strong predictive relationship between service type and attack likelihood. One-hot encoding would create 70 binary columns and lose this attack rate information. Additionally, analysis confirmed that R2L and U2R attacks predominantly utilize common services (ftp_data, ftp, telnet) with sufficient sample sizes (>400 records), ensuring smoothing introduced minimal signal loss (average 2.5% for R2L, 0.06% for U2R) and does not disadvantage minority class detection.

4.2.2.1. Protocol Type and Flag (14 categories total)

- `protocol_type` (3 categories: tcp, udp, icmp): One-hot encoded with drop_first=True → 2 binary features
- `flag` (11 categories: SF, S0, REJ, RSTR, etc.): One-hot encoded with drop_first=True → 10 binary features
- **Rationale:** Low-moderate cardinality makes one-hot encoding appropriate. These are nominal variables with no inherent ordering.

4.2.2.2. Target Variable Encoding

- `attack_category` (5 categories): Label encoded (Normal=1, DoS=0, Probe=2, R2L=3, U2R=4) for model compatibility
- Original categorical variable retained for interpretability in confusion matrices and error analysis

4.2.3. Outlier Detection and Handling

Systematic outlier detection using IQR method (outliers defined as values beyond Q1-1.5×IQR or Q3+1.5×IQR):

- 25 of 34 numerical features contained outliers
- 8 features showed >10% outliers

Feature	Outliers	% Affected	Max Value	Upper Bound
srv_diff_host_rate	28,399	22.5%	1.0	0.0
dst_host_same_src_port_rate	25,052	19.9%	1.0	0.15
dst_bytes	23,579	18.7%	1,309,937,401	1,290
dst_host_rerror_rate	22,795	18.1%	1.0	0.0
dst_host_srv_rerror_rate	19,357	15.4%	1.0	0.0
srv_rerror_rate	16,206	12.9%	1.0	0.0
rerror_rate	16,190	12.9%	1.0	0.0
src_bytes	13,840	11.0%	1,379,963,888	690

Decision: Outliers retained rather than removed

Rationale:

- In intrusion detection, extreme values often represent attacks themselves:
- Massive byte transfers (`src_bytes, dst_bytes`) indicate DoS attacks flooding network traffic
- High error rates signal reconnaissance and probing activities
- Extreme compromise/root counts indicate privilege escalation (U2R) attacks
- Removing outliers would eliminate the very attacks the models need to detect
- RobustScaler applied during feature scaling to handle extreme values without removal

4.2.4. Feature Scaling

Scaling method: RobustScaler (uses median and IQR, resistant to outliers)

Features scaled:

- Numerical features with extreme ranges: byte counts (`src_bytes, dst_bytes`), duration, connection counts (`count, srv_count`), compromise indicators (`num_compromised, num_root`)

Features excluded from scaling:

- Rate features (15 features): Already normalized to [0,1] range, scaling would distort interpretation
- One-hot encoded features (12 features): Binary indicators (0/1), scaling not appropriate
- Target-encoded service feature: Already normalized to [0,1] attack rate scale
- Target variable: `attack_category_encoded`

Data leakage prevention:

- Scaler fit on training data only
- Same transformation applied to test data using training parameters
- No information from test set used in scaling decisions

4.2.5. Feature Creation

Decision: No new engineered features created (no interaction terms, ratios, polynomial features, or binning)

Rationale:

- NSL-KDD features are already domain-engineered network statistics (connection rates, byte counts, error rates, temporal patterns)
- Project objectives focus on model comparison and identifying important existing features, not feature engineering innovation
- Maintaining interpretability for cybersecurity stakeholders—existing features (e.g., "`src_bytes`", "`logged_in`") are directly understandable
- Feature creation can be revisited in future iterations based on model performance and feature importance analysis

4.2.6. Final Feature Set

After feature engineering:

- Scaled numerical features: 19 (byte counts, connection counts, compromise indicators)
- Rate features (not scaled): 15 (already [0,1] range)
- One-hot encoded features: 12 (`protocol_type`: 2, `flag`: 10)
- Target-encoded feature: 1 (`service_encoded`)
- Total predictor features: 47
- Target variable: `attack_category_encoded` (5 classes: 0-4)

Dataset ready for exploratory data analysis and modelling

5. EDA and Discussion

5.1. Univariate Analysis

5.1.1. Numerical Features

The dataset contains 34 numerical variables. To maintain interpretability and focus on the most discriminative features, a subset was selected for detailed exploration based on:

ANOVA F-statistic ranking: Features were ranked by their ability to discriminate between the five attack categories (Normal, DoS, Probe, R2L, U2R). The F-statistic measures the ratio of between-group variance to within-group variance—higher values indicate stronger discrimination between attack types.

The top 9 features by F-statistic were selected for visualization (Figure 1), ranging from same_srv_rate (F=68,019) to count (F=21,428).

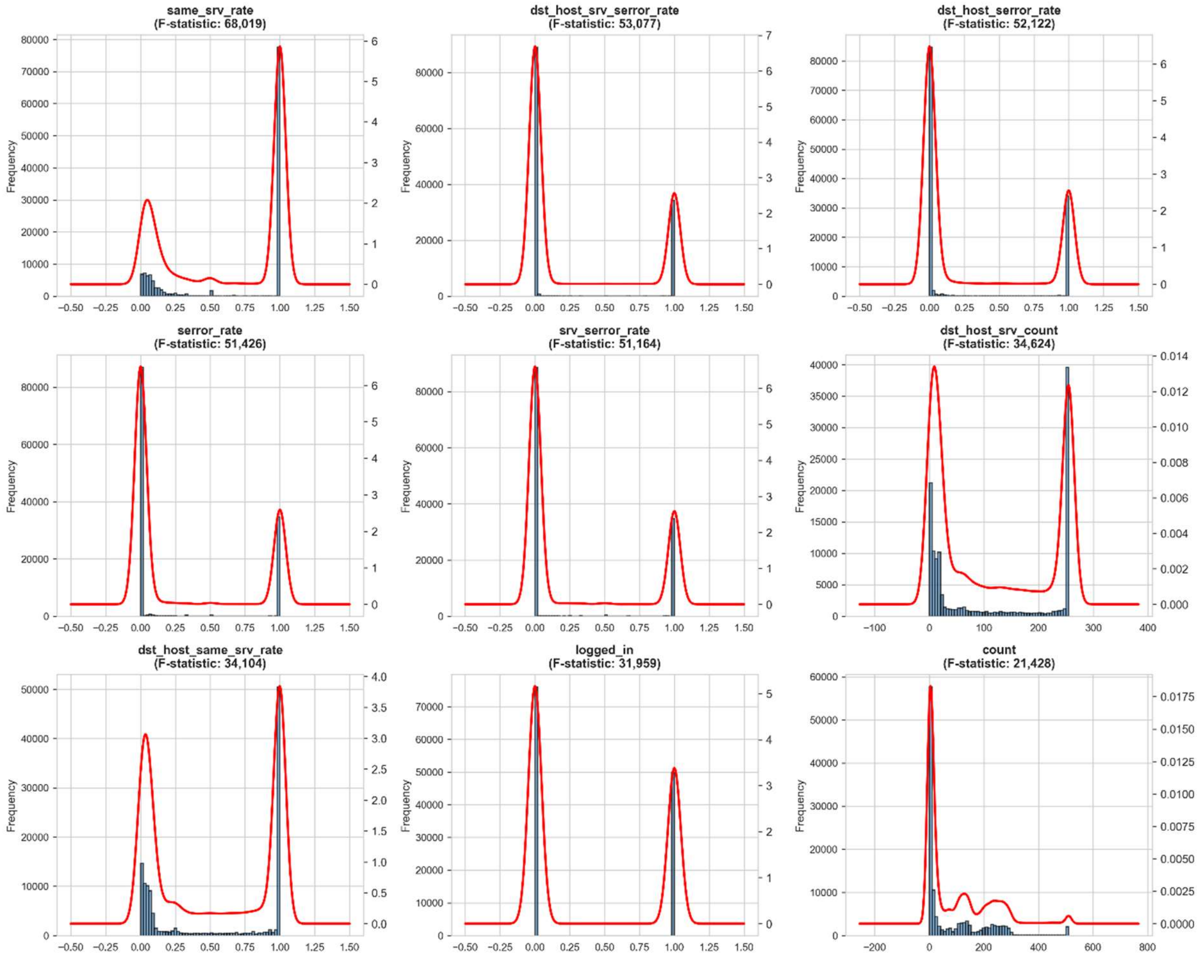


Figure 1: Distribution of top 9 numerical features by discriminative power. Histograms show frequency (left y-axis) with KDE overlay (right y-axis, red line). F-statistics indicate strength of discrimination between attack categories

5.1.1.1. Key distributional patterns:

- **Bimodal rate features:** The majority of rate features (`same_srv_rate, serror_rate, srv_serror_rate,` etc.) exhibit pronounced bimodal distributions with heavy concentration at 0.0 and 1.0. These features measure proportions: a value of 1.0 indicates all connections exhibited a behaviour (e.g., all had SYN errors, all targeted the same service), while 0.0 indicates none did. The bimodality reflects that network traffic tends toward "all or nothing" patterns—either a behaviour is consistently present or consistently absent—with intermediate values being rare. This bimodality is characteristic of network intrusion data where attacks produce systematic extreme patterns (DoS generating errors on all connections) while normal traffic shows the opposite (error-free connections).
- **Binary indicators:** The `logged_in` feature shows a clear binary distribution, reflecting whether a connection successfully logged into the system (1) or not (0). The heavy skew toward 0 indicates most connections in the dataset are not authenticated, which aligns with the prevalence of attack traffic**.**
- **Right-skewed count features:** Connection count variables (`count, dst_host_srv_count`) display strong right skewness with long tails. Most connections have low counts, but DoS

attacks generate extreme values (visible in the long right tails), creating natural discrimination between normal traffic and flooding attacks.

- **Zero-inflation in error rates:** Error rate features show substantial zero-inflation, where the majority of records have no errors (rate=0.0). Non-zero error rates appear as a secondary peak at 1.0, suggesting that when errors occur, they often affect all connections—a pattern consistent with reconnaissance and probing attacks.

These distributional characteristics directly inform modelling strategy: the bimodal nature of rate features suggests they are already well-suited for classification without transformation, while count features may benefit from the robust scaling already applied to handle extreme values.

5.1.2. Categorical Features

Four categorical features were analysed to understand the composition of network traffic types and attack distributions in the dataset.

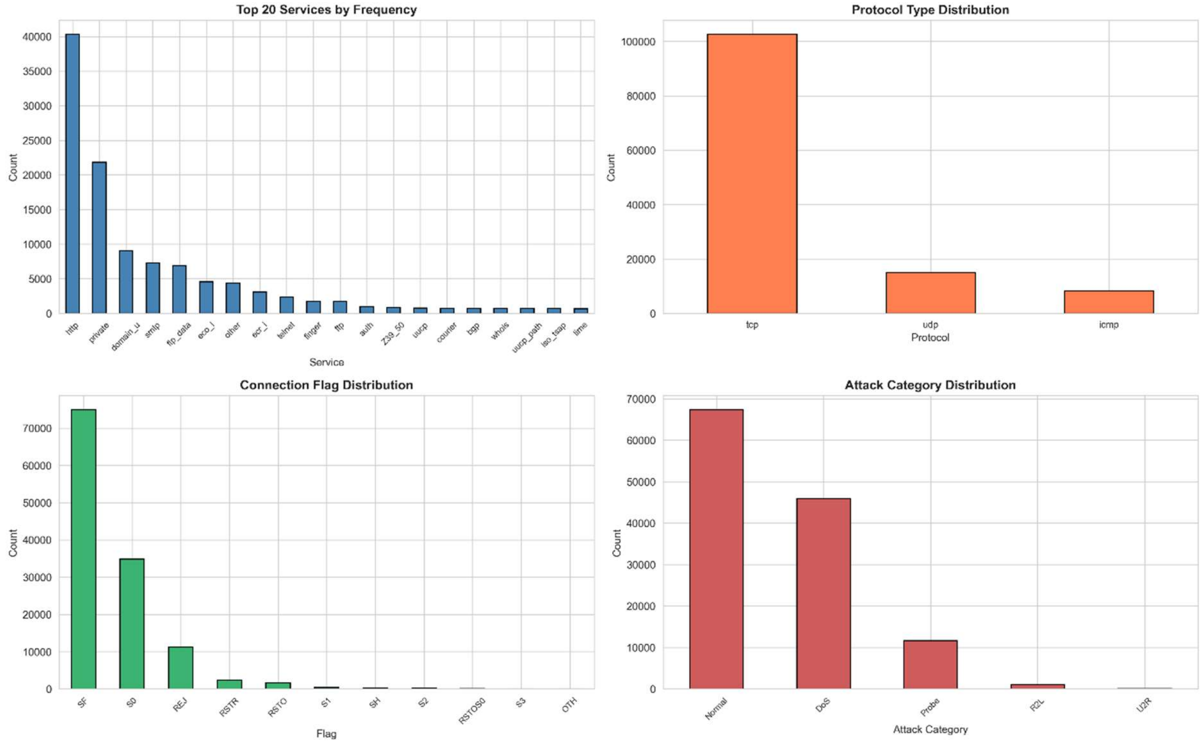


Figure 2: Distribution of categorical features. Top 20 services by frequency (top left), protocol type distribution (top right), connection flag distribution (bottom left), and attack category target variable (bottom right) showing severe class imbalance.

5.1.2.1. service (70 categories, top 20 shown)

The service feature exhibits a heavily skewed distribution dominated by http (40,338 records, 32% of the dataset) and private connections (21,853 records, 17%). This long-tail distribution is characteristic of network traffic data, where common protocols account for the majority of connections while specialized services appear infrequently. The top 10 services collectively represent approximately 80% of all traffic, with the remaining 60 services distributed across the tail. Given this high cardinality and the varying attack rates across services (ranging from 0% to 100% as identified in feature engineering), smoothed target encoding was applied to capture the predictive relationship between service type and attack likelihood while preventing overfitting on rare categories

5.1.2.2. protocol_type (3 categories)

Network protocol distribution aligns with expected internet traffic composition. TCP dominates with 103,599 records (82% of connections), reflecting its role as the primary transport protocol for most internet applications. UDP accounts for 16,347 records (13%), primarily supporting connectionless services like DNS queries and streaming applications. ICMP represents 6,027 records (5%), typically used for network diagnostics and control messages. This distribution is consistent with typical network monitoring datasets and provides meaningful categorical separation for classification modelling. The feature was one-hot encoded to create binary indicator variables for each protocol type.

5.1.2.3. flag (11 categories):

Connection state flags reveal patterns in how network connections are established and terminated. SF (successful connection) is the most prevalent flag with 76,813 records (61%), indicating properly established and closed connections. S0 (connection attempt with no response) appears in 35,026 records (28%). This flag indicates incomplete connections where SYN packets were sent but never acknowledged—a characteristic pattern of SYN flood DoS attacks where attackers intentionally leave connections in half-open states to exhaust server resources. This flag shows a relatively balanced distribution across attack types, indicating connection rejections occur through multiple mechanisms: security policies blocking malicious traffic, legitimate connection failures, and firewall responses to various attack attempts.

RSTR (connection reset by responder) represents approximately 1% of connections but serves as a strong probe indicator—when attackers scan closed ports during reconnaissance, target hosts typically respond by resetting the connection.

The remaining seven flag types (RSTO, S1, S2, S3, SH, OTH) are comparatively rare, each representing less than 1% of connections. This distribution suggests that most network traffic falls into clear categories of either successfully established connections, failed DoS attempts, or rejected connections, with intermediate or error states being uncommon. The flag variable was one-hot encoded with drop_first=True to avoid multicollinearity in subsequent modelling.

5.1.2.4. attack_category (5 categories, target variable):

The target variable demonstrates severe class imbalance, a critical consideration for model development and evaluation. Normal traffic represents 67,343 records (53.5% of the dataset), while DoS attacks account for 45,927 records (36.5%). These two categories comprise nearly 90% of all observations. Probe attacks, representing reconnaissance and scanning activities, contribute 11,656 records (9.3% of the dataset). The minority classes—R2L (Remote to Local attacks, 995 records, 0.8%) and U2R (User to Root privilege escalation, 52 records, 0.04%)—are severely underrepresented despite representing critical security threats. The extreme rarity of U2R attacks, with only 52 training examples, presents a substantial challenge for classification models. This class imbalance necessitates specialised handling during the modelling phase, including techniques such as SMOTE (Synthetic Minority Over-sampling Technique) or class weighting, to ensure the models can adequately detect these rare but high-impact attack categories rather than simply predicting the majority classes.

5.2. Bivariate Analysis

Bivariate analysis examined feature relationships and discriminative power across attack categories. Predictor relationships were assessed through correlation analysis (numerical features) and Cramér's V (categorical features) to identify multicollinearity. Feature discrimination was evaluated through distribution analysis (numerical) and categorical associations with attack types

5.2.1. Feature Relationships (predictor-predictor associations)

5.2.1.1. Numerical Feature Correlations

5.2.1.1.1. Correlation Analysis and Multicollinearity Detection

A correlation heatmap was constructed for all 34 numerical features to identify redundant information and potential multicollinearity issues that could impact model performance. The analysis revealed several clusters of highly correlated features (|r| > 0.8), indicating substantial redundancy in certain feature groups.

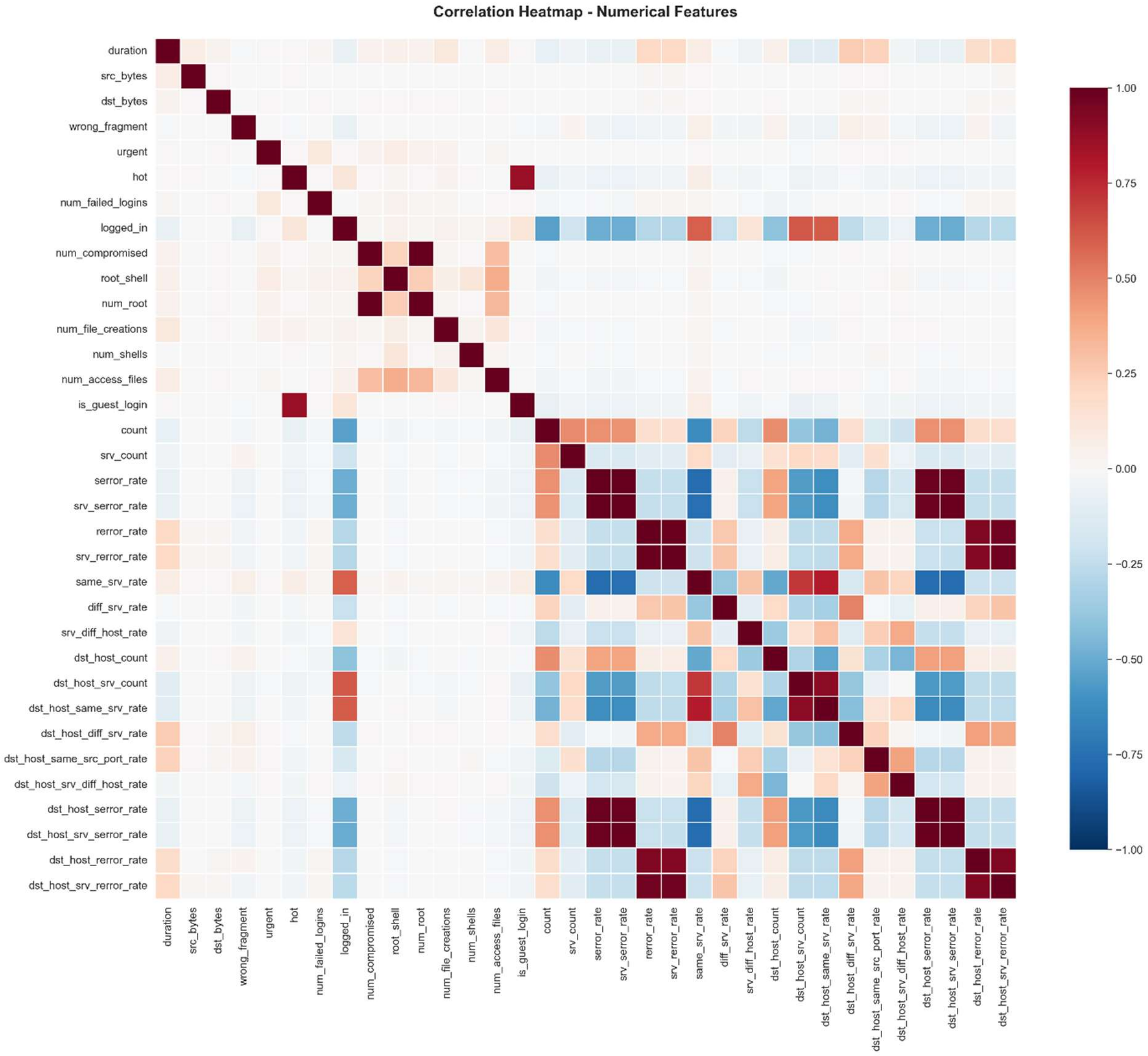


Figure 3: Correlation heatmap of numerical features. Red indicates positive correlation, blue indicates negative correlation. Several distinct clusters of highly correlated features are visible, particularly among error rate and connection state variables.

5.2.1.1.2. Highly correlated feature groups identified

- **Compromise indicators (r = 0.999):** The features `num_compromised` and `num_root` demonstrate near-perfect correlation, reflecting the sequential nature of privilege escalation attacks. Attackers first gain access to system resources (incrementing num_compromised), then attempt to escalate to full administrator control (incrementing num_root). This progression is characteristic of U2R (User-to-Root) attacks where gaining complete system control is the primary objective.
- **SYN error rate cluster (r > 0.97):** A strongly correlated group emerged among `serror_rate, srv_serror_rate, dst_host_serror_rate`, and `dst_host_srv_serror_rate`. These features measure connection errors at progressively broader network contexts—from individual connections to service-level, host-level, and host-service-level aggregations. The high correlations (ranging from 0.977 to 0.993) indicate that SYN errors tend to propagate consistently across these hierarchical levels, suggesting that when connection errors occur, they affect multiple network layers simultaneously. This pattern is characteristic of DoS attacks where flooding generates systematic errors across the entire connection infrastructure.
- **REJ error rate cluster (r > 0.92):** Similarly, `rerror_rate, srv_rerror_rate, dst_host_rerror_rate`, and `dst_host_srv_rerror_rate` form a correlated cluster representing rejected connection attempts at various aggregation levels. The slightly lower correlations (0.917 to 0.989 compared to the serror cluster) suggest greater variability in how connection rejections propagate through the network, potentially reflecting different defensive responses to various attack types.
- **Service volume relationship (r = 0.897):** The strong correlation between `dst_host_srv_count` and `dst_host_same_srv_rate` indicates that as the number of connections to a specific service increases, the proportion of connections to that same service also increases—a logical relationship in network traffic patterns where popular services naturally accumulate more connections.

Activity indicators (r = 0.860): The correlation between `hot` (indicators of sensitive file/directory access) and `is_guest_login` suggests that guest login sessions disproportionately trigger access to sensitive system resources, potentially indicating reconnaissance or privilege escalation attempts.

5.2.1.1.3. Implications for modelling

While this multicollinearity is substantial, its impact varies by model type. Tree-based ensemble methods (Random Forest, XGBoost) handle correlated features effectively through their splitting mechanisms and can extract complementary information even from highly correlated variables. Logistic Regression may be more sensitive to multicollinearity, potentially inflating coefficient standard errors. However, given that these correlated features measure conceptually related but distinct network behaviours (e.g., errors at connection vs. host vs. service level), retaining all features initially allows models to learn which aggregation level provides the strongest signal for each attack type. Feature importance analysis following model training will inform any necessary feature reduction.

5.2.1.2. Categorical Predictor Relationships

Beyond numerical features, categorical predictors were examined for mutual associations using Cramér's V, a measure of association strength for nominal variables (ranging from 0=no association to 1=perfect association).

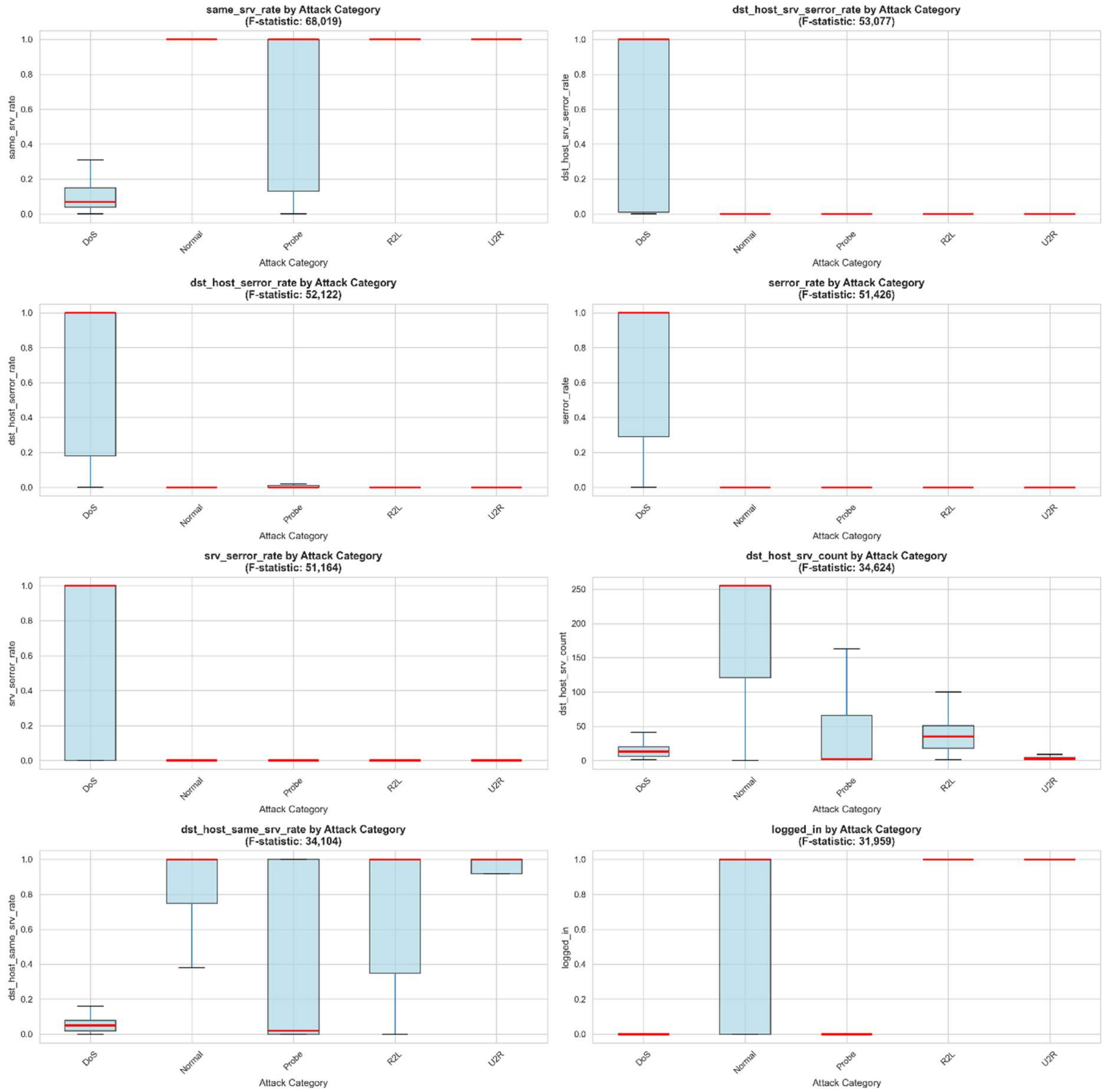
Category combination	V	Association
protocol_type vs service:	0.923	very strong
protocol_type vs flag:	0.278	weak-moderate
service vs flag:	0.299	weak-moderate

The strong protocol-service association (V=0.923) reflects expected network architecture—services are largely protocol-bound (e.g., HTTP operates on TCP, DNS on UDP). However, this correlation does not create redundancy in the final feature set because the two variables are encoded using different strategies that capture complementary information: protocol_type (one-hot encoded) preserves protocol identity, while service (target-encoded) captures attack risk per service. The moderate associations between flag and other categorical features (V<0.3) indicate sufficient independence. All categorical features are retained as they provide complementary predictive signals for tree-based models.

5.2.2. Feature Discrimination by Attack Category (predictor-target)

5.2.2.1. Numerical Features by Attack Category

To understand how features discriminate between Normal traffic and the four attack categories (DoS, Probe, R2L, U2R), the top eight features by F-statistic were visualized using boxplots grouped by attack category. This analysis reveals the distinct behavioural signatures that characterize each attack type.


same_srv_rate by Attack Category (F-statistic: 68,019)

dst_host_srv_serror_rate by Attack Category (F-statistic: 53,077)

dst_host_serror_rate by Attack Category (F-statistic: 52,122)

serror_rate by Attack Category (F-statistic: 51,426)

srv_serror_rate by Attack Category (F-statistic: 51,164)

dst_host_srv_count by Attack Category (F-statistic: 34,624)

dst_host_same_srv_rate by Attack Category (F-statistic: 34,104)

logged_in by Attack Category (F-statistic: 31,959)

Figure 4: Distribution of top 8 features across attack categories. Boxplots show median (red line), interquartile range (box), and range (whiskers). Features demonstrate clear separation between attack types, validating their discriminative power indicated by high F-statistics.

5.2.2.1.1. Error rate features as DoS signatures:

The four error rate features—`serror_rate`, `srv_serror_rate`, `dst_host_serror_rate`, and `dst_host_srv_serror_rate`—exhibit remarkably consistent discrimination patterns with near-perfect separation between DoS attacks and all other categories. DoS attacks demonstrate mean error rates of approximately 0.748 across all error metrics, indicating that roughly 75% of connections in DoS attacks generate SYN errors. In stark contrast, Normal traffic maintains mean error rates of only 0.012-0.014, while Probe, R2L, and U2R attacks show similarly low error rates (<0.05). The boxplots reveal that DoS attacks display substantial internal variability (boxes spanning from approximately 0.2 to 1.0), reflecting different DoS attack subtypes with varying error generation patterns. All non-DoS categories show minimal variability with distributions tightly clustered near zero. This distinctive error signature makes these features extremely powerful DoS detectors, explaining their exceptionally high F-statistics (>51,000).

5.2.2.1.2. same_srv_rate reveals attack-specific connection patterns

The same_srv_rate feature (proportion of connections to the same service) demonstrates distinct behavioural profiles across attack categories. DoS attacks exhibit the lowest mean (0.192), reflecting their strategy of generating diverse connection attempts to overwhelm network resources rather than targeting a single service repeatedly. This contrasts sharply with Normal traffic (mean 0.969), where legitimate users naturally establish multiple connections to the same services (e.g., web browsers making repeated HTTP requests). Probe attacks show intermediate values (mean 0.697), consistent with reconnaissance activities that scan multiple services but may focus scanning efforts on specific targets. Most notably, R2L and U2R attacks display very high same_srv_rate values (0.997 and 0.932 respectively), indicating highly focused attacks repeatedly targeting specific vulnerable services to gain unauthorized access or escalate privileges. The clear separation in medians across categories (visible as distinct red lines in the boxplot) demonstrates why this feature achieves the highest F-statistic (68,019) in the dataset.

5.2.2.1.3. dst_host_srv_count distinguishes attack intensities

Connection volume to specific services (dst_host_srv_count) provides strong discrimination between Normal traffic and attacks. Normal traffic exhibits a mean of 190.286 connections, reflecting the high-volume, sustained nature of legitimate network activity. DoS attacks show substantially reduced counts (mean 26.524), approximately seven times lower than Normal traffic, because DoS floods create many brief, failed connection attempts rather than sustained successful connections. Probe attacks demonstrate intermediate volumes (mean 42.367), consistent with systematic but targeted scanning activities. U2R attacks show the lowest connection counts (mean 9.885), nearly 20 times lower than Normal traffic, reflecting their highly focused nature—these attacks require only a few successful connections to achieve privilege escalation objectives. The boxplots reveal considerable variability within categories, particularly for Normal traffic, indicating diverse legitimate usage patterns.

5.2.2.1.4. logged_in identifies authentication patterns

The logged_in binary indicator reveals critical differences in authentication requirements across attack categories. Normal traffic and U2R attacks both show high authentication rates (means near 1.0), but for fundamentally different reasons. Normal traffic represents legitimate user sessions where authentication is standard procedure. U2R attacks require authentication because their objective—privilege escalation—necessitates first gaining legitimate user access before attempting to escalate to root privileges. In contrast, DoS, Probe, and R2L attacks show authentication rates near zero (means <0.05), as these attack types operate at the network level without requiring authenticated sessions. DoS attacks aim to exhaust resources through connection floods, Probe attacks perform reconnaissance through unauthenticated scans, and R2L attacks exploit vulnerabilities to gain initial unauthorized access from remote locations. This authentication dichotomy provides a clear discriminative signal, particularly for distinguishing U2R attacks (which superficially resemble Normal traffic in many features) from other attack categories.

5.2.2.2. Categorical Features by Attack Category

To complement the numerical feature analysis, categorical variables were examined for their associations with attack types using crosstabulation analysis. This reveals which network protocols and connection states are preferentially associated with specific attack categories.

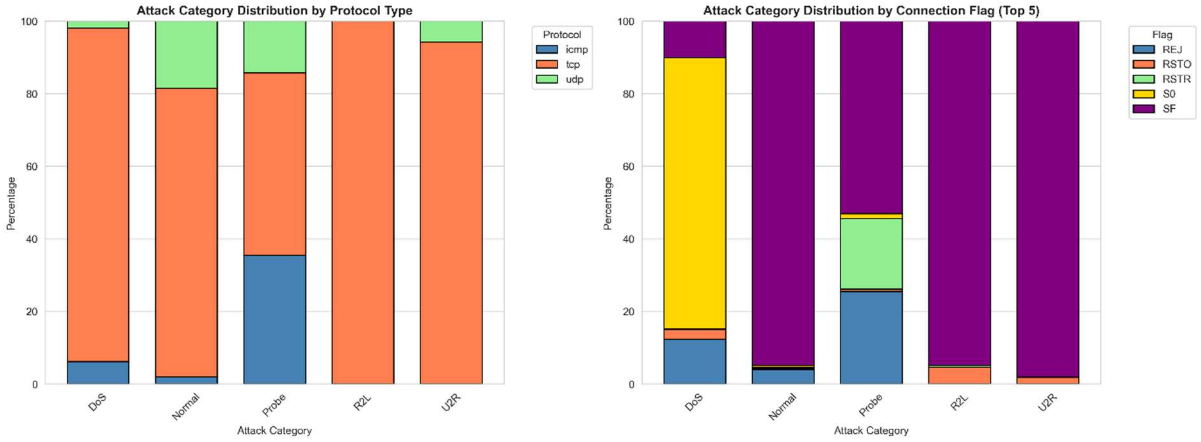


Figure 5: Stacked bar charts showing attack category composition by protocol type (left) and connection flag (right). Each bar represents 100% of that attack category, with colours indicating the proportion using each protocol or flag type.

5.2.2.2.1. Protocol type preferences across attack categories

The relationship between network protocol and attack type reveals distinct protocol preferences driven by the technical requirements of different attack strategies.

- **TCP dominates** the attack landscape, accounting for 91.9% of DoS attacks, 100% of R2L attacks, and 94.2% of U2R attacks. This TCP preference reflects fundamental requirements: DoS attacks commonly employ TCP SYN flooding to exhaust server resources, while R2L and U2R attacks require the reliable, connection-oriented communication that TCP provides to exploit application vulnerabilities and establish authenticated sessions for privilege escalation.
- **ICMP demonstrates a striking association with reconnaissance activities**. While ICMP represents only 6.6% of total traffic, it accounts for 35.5% of all Probe attacks. This pattern reflects ICMP's role in network diagnostics: attackers leverage ICMP echo requests (ping) for host discovery and network mapping during reconnaissance phases. The absence of R2L and U2R attacks in ICMP traffic is expected, as ICMP's limited functionality cannot support the application-layer exploitation required for these attack types.
- **UDP exhibits the most benign profile among protocols.** UDP accounts for only 1.9% of DoS attacks and is absent from R2L attacks entirely, indicating minimal use by attackers. In contrast, UDP represents a more substantial portion of Normal traffic (18.5%), reflecting UDP's primary use for legitimate connectionless services such as DNS queries, streaming media, and VoIP. The protocol's stateless nature makes it less suitable for systematic flooding patterns characteristic of DoS attacks or the exploitation-based strategies required for R2L and U2R attacks.

5.2.2.2.2. Connection state flags as attack indicators:

Connection state flags provide powerful discriminative signals, with certain flag types serving as near-definitive indicators of specific attack categories.

- **The S0 flag (connection attempt with no response) emerges as the strongest DoS signature**: 98.5% of all S0 connections are DoS attacks, and S0 flags account for 74.8% of all DoS traffic. This overwhelming association reflects the technical mechanism of TCP SYN flood attacks, where attackers send connection initiation packets (SYN) but never complete the three-way handshake, leaving connections in the S0 state and exhausting server resources through accumulated half-open connections.
- **The RSTR flag (connection reset by responder) demonstrates an equally strong association with Probe attacks**: 90.0% of RSTR connections are Probes. This pattern emerges from target systems responding to port scans—when attackers attempt connections to closed ports during reconnaissance, target hosts typically respond by resetting the connection. The high RSTR proportion in Probe traffic (19.4% of all Probes) indicates that a substantial portion of scanning activity targets closed or filtered ports, generating characteristic reset responses.
- **The SF flag (successful connection, properly closed) predominates in Normal traffic**, accounting for 94.9% of legitimate connections. This reflects standard protocol behaviour where connections are established, data is exchanged, and connections are properly terminated. However, SF also appears in 13.0% of Probe attacks, indicating that sophisticated reconnaissance can mimic legitimate connection patterns. The presence of SF in 94.4% of R2L and 98.1% of U2R attacks is particularly notable—these attack types require successful connection establishment to exploit application vulnerabilities or perform privilege escalation, distinguishing them from the connection-disrupting patterns of DoS attacks.
- **The REJ flag (connection rejected) shows a relatively balanced distribution** across DoS (50.5%), Normal (24.0%), and Probe (25.5%) categories. This distribution suggests that connection rejections occur through multiple mechanisms: security policies blocking malicious traffic, legitimate connection failures, and firewall responses to scanning attempts.
- **The RSTO flag (connection reset by originator) shows strong association with DoS** (77.8%), representing scenarios where attackers initiate then immediately terminate connections to maximize disruption with minimal resource commitment.

6. Key Insights

The exploratory data analysis reveals a well-structured dataset with clear discriminative patterns, while also identifying specific challenges that will require careful handling during model development.

6.1. Data characteristics

The NSL-KDD dataset demonstrates high quality (no missing values, no duplicates) but exhibits severe class imbalance with U2R (52 records, 0.04%) and R2L (995 records, 0.8%) representing less than 1% of data. Outliers in byte counts and connection features were retained as they represent actual attacks rather than noise. RobustScaler was applied to normalize features while preserving attack discrimination.

6.2. Feature patterns

Rate features exhibit natural bimodality (values concentrated at 0.0 and 1.0) reflecting "all or nothing" network behaviour, requiring no transformation for classification. Connection count features show right skewness with extreme values indicating attacks. Numerical features display substantial multicollinearity (error rate clusters r > 0.97, compromise indicators r = 0.999), which poses minimal concern for tree-based models but may require regularization for Logistic Regression.

6.3. Discriminative power

Features provide exceptionally clear attack signatures: DoS attacks show extreme error rates (0.75 vs 0.01 Normal) and S0 flags (98.5% DoS); Probe attacks exhibit RSTR flags (90% Probe) and ICMP protocol usage (49.9% of ICMP is Probe); R2L/U2R attacks display focused targeting (`same_srv_rate` > 0.93) and service-specific patterns; Normal traffic maintains consistent authenticated behaviour with low error rates. This clear separation suggests high classification performance is achievable even with relatively simple algorithms.

6.4. Feature engineering validation

Smoothed target encoding successfully captured service attack rates while introducing minimal signal loss for minority classes (2.5% for R2L, 0.06% for U2R). One-hot encoding of categorical variables preserved strong discriminative signatures. The hierarchical feature structure (connection → service → host → host-service levels) enables detection of attack patterns at different network scopes.

6.5. Modelling readiness

The 47-feature dataset provides strong discriminative signals with complementary numerical (continuous patterns) and categorical (mechanism indicators) information. However, extreme class imbalance remains the critical challenge—specialised techniques such as SMOTE or class weighting will be essential to ensure models detect rare U2R and R2L attacks rather than defaulting to majority class prediction.

7. Classification Modelling Analysis

7.1. Logistic Regression (Baseline)

7.1.1. Model Configuration

Logistic Regression was selected as the baseline model to establish initial performance benchmarks and provide a computationally efficient linear classifier for comparison against more complex tree-based approaches. The model was configured with maximum iterations set to 1000 to ensure convergence on this multi-class classification problem, using the default L2 regularization and one-vs-rest strategy for handling the five attack categories.

7.1.2. Overall Performance

The table below presents the primary performance metrics for the Logistic Regression baseline model.

Metric	Value
Macro Avg Recall	0.4011
Macro Avg F1-Score	0.3972
ROC-AUC (Macro)	0.7432
Overall Accuracy	0.6832

While accuracy appears moderate at 68.32%, this metric is misleading given the severe class imbalance. The model achieves high accuracy primarily by correctly classifying the abundant Normal and DoS classes while failing almost entirely on minority classes.

7.1.3. Performance by Attack Category

The following table provides detailed per-class performance metrics, revealing dramatic performance variation across attack categories:

Attack Category	Precision	Recall	F1-Score	Support
DoS	0.8017	0.7274	0.7627	7458
Normal	0.6198	0.9443	0.7484	9711
Probe	0.8228	0.3337	0.4749	2421
R2L	0.0000	0.0000	0.0000	2754
U2R	0.0000	0.0000	0.0000	200

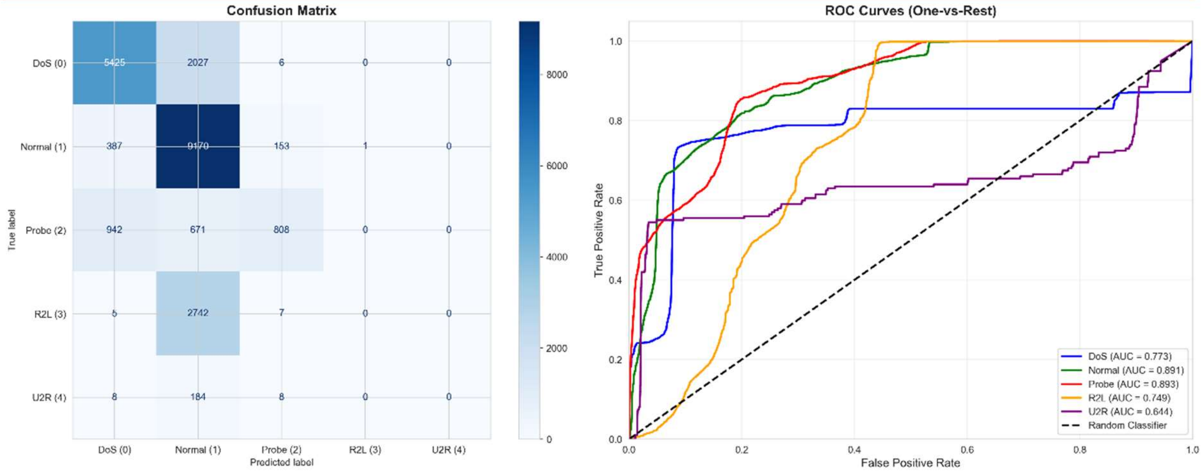


Figure 6: Logistic Regression Evaluation Results. Left: Confusion matrix showing actual versus predicted classifications, revealing that 2,742 R2L attacks and 184 U2R attacks were misclassified as Normal traffic. Right: ROC curves (one-vs-rest) demonstrating discriminative ability per class, with AUC scores ranging from 0.644 (U2R) to 0.893 (Probe).

7.1.3.1. Majority Classes (Good Performance):

- DoS attacks: 72.74% recall, demonstrating adequate detection of denial-of-service patterns
- Normal traffic: 94.43% recall, correctly identifying legitimate network behaviour

7.1.3.2. Probe Attacks (Poor Performance):

- Recall: 33.37%, missing two-thirds of reconnaissance attempts
- Examination of the confusion matrix (Figure 6) shows 942 Probe attacks misclassified as DoS and 671 as Normal

7.1.3.3. Minority Classes (Complete Failure):

- R2L attacks: 0% recall - failed to predict this class entirely
  - Confusion matrix reveals 2,742 of 2,754 R2L attacks misclassified as Normal traffic
- U2R attacks: 0% recall - failed to predict this class entirely
  - 184 of 200 U2R attacks misclassified as Normal traffic

This pattern demonstrates the classic class imbalance problem where the model defaults to predicting majority classes to maximize overall accuracy at the expense of minority class detection.

7.1.4. ROC-AUC Analysis

Figure 6 presents the confusion matrix and ROC curves for the Logistic Regression model. Despite zero recall for minority classes, the ROC curves reveal that the model does possess discriminative ability:

Attack Category	ROC-AUC	Interpretation
Probe	0.893	Strong discrimination
Normal	0.891	Strong discrimination
DoS	0.773	Moderate discrimination
R2L	0.749	Moderate discrimination despite 0% recall
U2R	0.644	Weak discrimination barely above random (0.5)

The disconnect between AUC scores and recall indicates that the model assigns different probability scores to classes but sets decision thresholds too conservatively to actually predict minority classes. The R2L AUC of 0.749 suggests the model has learned some discriminative patterns but lacks sufficient confidence to overcome the class imbalance bias during prediction. The U2R AUC of 0.644, barely above random classification, indicates the model struggles fundamentally to distinguish these rare privilege escalation attacks.

7.2. Random Forest

7.2.1. Model Configuration

Random Forest was selected as the second model to evaluate whether ensemble tree-based methods could better handle the non-linear attack signatures and class imbalance observed with Logistic Regression. The Random Forest Classifier was configured with 100 decision trees (`n_estimators`=100) as a baseline setting, using bootstrap sampling and feature randomization to create diverse trees within the ensemble. All other hyperparameters remained at default values to establish untuned baseline performance for fair comparison with subsequent models.

7.2.2. Overall Performance

The table below presents the primary performance metrics for the Random Forest model.

Metric	Value
Macro Avg Recall	0.5067
Macro Avg F1-Score	0.5081
ROC-AUC (Macro)	0.8908
Overall Accuracy	0.7654

All key metrics show substantial improvement over Logistic Regression, with macro-averaged recall increasing from 0.4011 to 0.5067 (26% improvement) and ROC-AUC improving from 0.7432 to 0.8908.

7.2.3. Performance by Attack Category

The following table provides detailed per-class performance metrics, revealing dramatic performance variation across attack categories:

Attack Category	Precision	Recall	F1-Score	Support
DoS	0.9601	0.8034	0.8748	7458
Normal	0.6627	0.9737	0.7887	9711
Probe	0.8857	0.7109	0.7887	2421
R2L	0.9655	0.0305	0.0591	2754
U2R	0.6000	0.0150	0.0293	200

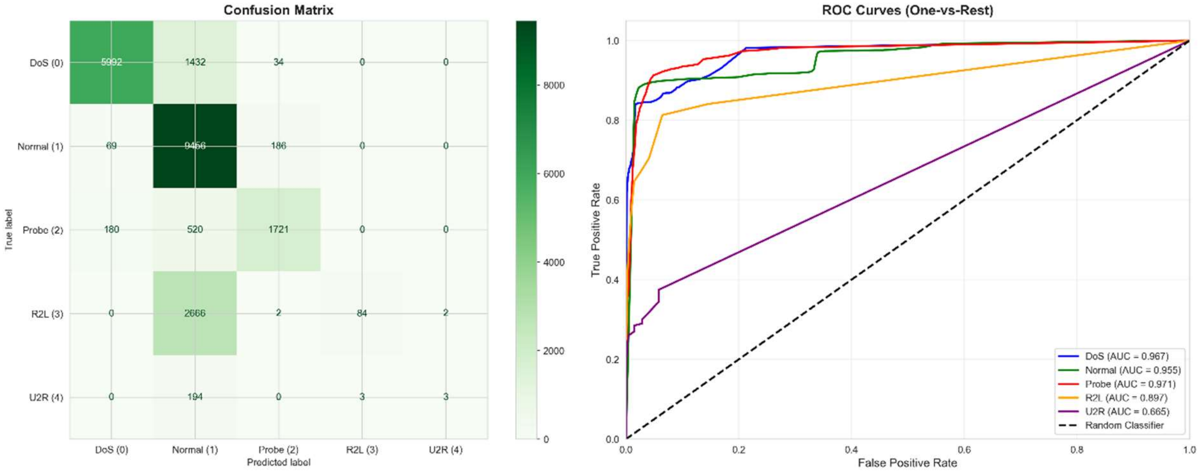


Figure 7: Random Forest Evaluation Results. Left: Confusion matrix revealing that 2,666 R2L attacks and 194 U2R attacks were misclassified as Normal traffic. Right: ROC curves (one-vs-rest) with AUC scores ranging from 0.665 (U2R) to 0.971 (Probe).

7.2.3.1. Majority Classes (Good Performance):

- DoS attacks: 80.34% recall, demonstrating improved detection of denial-of-service patterns
- Normal traffic: 97.37% recall, continuing to correctly identify legitimate network behaviour

7.2.3.2. Probe Attacks (Medium Performance):

- Recall: 71.09%, substantially improved detection

- Examination of the confusion matrix (Figure 7) shows the number of Probe attacks misclassified as DoS has dropped significantly to 180 and Probe attacks misclassified as Normal reduced slightly to 520

7.2.3.3. Minority Classes (Improved but still poor):

- R2L attacks: 3.05% recall – a small improvement on the logistic regression model
    - However, Confusion matrix reveals 2,666 of 2,754 R2L attacks still misclassified as Normal traffic
- U2R attacks: 1.50% recall - a slight improvement on the logistic regression model
    - 194 of 200 U2R attacks misclassified as Normal traffic, with only 3 attacks correctly identified)

Despite these improvements this model continues to demonstrate class imbalance, defaulting to predicting majority classes to maximize overall accuracy at the expense of minority class detection.

7.2.4. ROC-AUC Analysis

Figure 7 presents the confusion matrix and ROC curves for the Random Forest model. Despite low recall for minority classes, the ROC curves reveal that the model demonstrates improved discriminative ability:

Attack Category	ROC-AUC	Interpretation
Probe	0.971	Very strong discrimination
Normal	0.955	Very strong discrimination
DoS	0.967	Strong discrimination
R2L	0.897	Moderate discrimination despite low recall
U2R	0.665	Moderate discrimination despite low recall

The ROC-AUC scores demonstrate substantially better discrimination than Logistic Regression, with Probe attack detection improving dramatically—recall more than doubled from 33.37% to 71.09%. R2L discrimination also strengthened (AUC: 0.749 → 0.897), indicating tree-based ensembles effectively capture complex attack signatures. However, the disconnect between high AUC scores and extremely low recall for minority classes (R2L: 3.05%, U2R: 1.50%) reveals that severe class imbalance continues to bias predictions toward majority classes. Random Forest represents significant advancement over the linear baseline, yet minority class detection remains unresolved without explicit class imbalance mitigation strategies.

7.3. XGBoost

7.3.1. Model Configuration

XGBoost was selected as the third model to evaluate whether gradient boosting could achieve further improvements over Random Forest's ensemble approach. Unlike Random Forest which builds trees independently, XGBoost constructs trees sequentially, with each new tree correcting errors made by previous trees. The XGBClassifier was configured with 100 boosting rounds (`n_estimators`=100), learning rate of 0.1, and maximum tree depth of 6, matching Random

Forest's tree count for fair baseline comparison. All other hyperparameters remained at default values to establish untuned baseline performance.

7.3.2. Overall Performance

The table below presents the primary performance metrics for the XGBoost model.

Metric	Value
Macro Avg Recall	0.5210
Macro Avg F1-Score	0.5260
ROC-AUC (Macro)	0.9432
Overall Accuracy	0.7580

XGBoost demonstrates incremental improvements over Random Forest on primary metrics, with macro-averaged recall increasing from 0.5067 to 0.5210 and ROC-AUC improving from 0.8908 to 0.9432. Notably, overall accuracy decreased slightly from 76.54% to 75.80%, reflecting the appropriate trade-off of majority class recall for improved minority class detection. This inverse relationship between accuracy and minority class recall reinforces that accuracy is an inappropriate primary metric for imbalanced datasets—XGBoost achieves the strongest discriminative ability despite lower overall accuracy.

7.3.3. Performance by Attack Category

The following table provides detailed per-class performance metrics, revealing dramatic performance variation across attack categories:

Attack Category	Precision	Recall	F1-Score	Support
DoS	0.8925	0.8162	0.8526	7458
Normal	0.6707	0.9226	0.7767	9711
Probe	0.8533	0.7617	0.8049	2421
R2L	0.9795	0.0694	0.1295	2754
U2R	0.6364	0.0350	0.0664	200

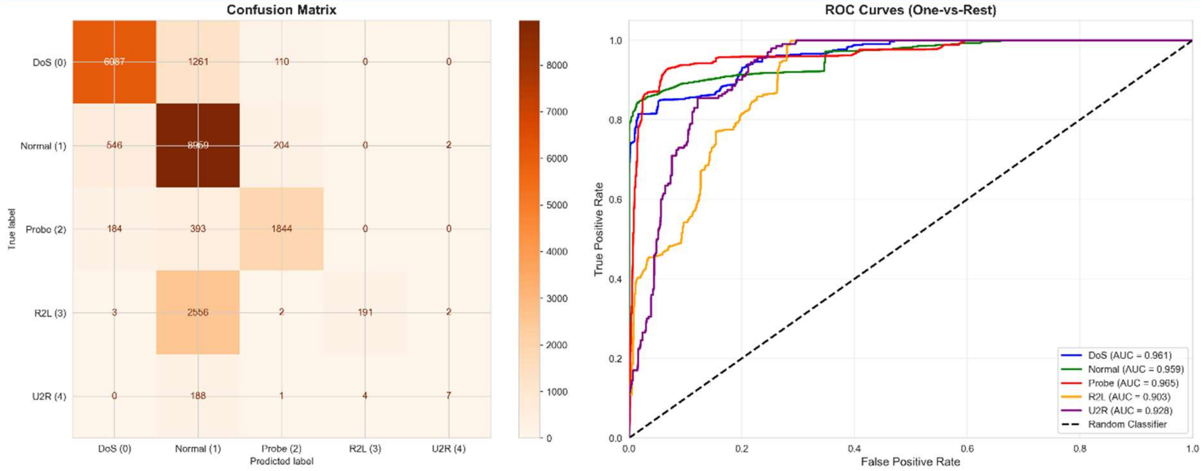


Figure 8: XGBoost Evaluation Results. Left: Confusion matrix revealing that 2,556 R2L attacks and 188 U2R attacks were misclassified as Normal traffic. Right: ROC curves (one-vs-rest) with AUC scores ranging from 0.903 (R2L) to 0.965 (Probe).

7.3.3.1. Majority Classes (Good Performance):

- DoS attacks: 81.62% recall, maintaining strong detection of denial-of-service patterns
- Normal traffic: 92.26% recall, continuing to correctly identify legitimate network behaviour

7.3.3.2. Probe Attacks (Good Performance):

- Recall: 76.17%, achieving the highest Probe detection of all three models
- Examination of the confusion matrix shows 184 Probe attacks misclassified as DoS and 393 as Normal, representing continued improvement in reconnaissance detection

7.3.3.3. Minority Classes (Improved but still poor):

- R2L attacks: 6.94% recall – more than doubled from Random Forest's 3.05%
  - Confusion matrix reveals 191 attacks correctly identified (vs 84 for Random Forest)
  - However, 2,556 of 2,754 R2L attacks still misclassified as Normal traffic

- U2R attacks: 3.50% recall – more than doubled from Random Forest's 1.50%

  - 7 attacks correctly identified (vs 3 for Random Forest), with 188 misclassified as Normal

XGBoost shows measurable improvement on minority classes through its sequential error-correction mechanism, yet severe class imbalance continues to dominate prediction behaviour.

7.3.4. ROC-AUC Analysis

Figure 8 presents the confusion matrix and ROC curves for the XGBoost model. The ROC curves demonstrate exceptional discriminative ability across all attack categories:

Attack Category	ROC-AUC	Interpretation
Probe	0.965	Excellent discrimination
DoS	0.961	Excellent discrimination
Normal	0.959	Excellent discrimination
U2R	0.928	Very strong discrimination
R2L	0.903	Very strong discrimination

XGBoost achieves the strongest discrimination of all three models, with particularly dramatic improvement for U2R attacks (AUC: 0.665 → 0.928), demonstrating that gradient boosting effectively learns patterns for even the rarest attack type. All classes now exceed 0.90 AUC, indicating excellent separability in the probability space. However, the persistent disconnect between near-perfect AUC scores and extremely low recall for minority classes (R2L: 6.94%, U2R: 3.50%) confirms that class imbalance—not lack of discriminative patterns—remains the fundamental barrier to effective detection. While XGBoost represents the best baseline performance, minority class detection requires explicit class imbalance mitigation strategies such as SMOTE or class weighting to translate strong discrimination into actionable predictions.

7.4. Comparative Performance Across Models

The table below presents a side-by-side comparison of the three baseline models across primary evaluation metrics, revealing clear performance progression from linear to tree-based approaches.

Model	Macro Avg Recall	Macro Avg F1	ROC-AUC (Macro)	Accuracy
Logistic Regression	0.4011	0.3972	0.7432	0.6832
Random Forest	0.5067	0.5081	0.8908	0.7654
XGBoost	0.5210	0.5260	0.9432	0.7580

The progression demonstrates consistent improvement on cybersecurity-critical metrics: macro-averaged recall increased 30% from Logistic Regression to XGBoost (0.4011 → 0.5210), while ROC-AUC improved 27% (0.7432 → 0.9432). The slight decrease in accuracy from Random Forest to XGBoost (76.54% → 75.80%) reflects the appropriate trade-off where the model predicts minority classes more frequently, reducing majority class recall but improving detection of critical R2L and U2R attacks.

7.4.1. Minority Class Performance Comparison

The following table examines recall performance specifically for the critical minority attack classes that represent the most severe cybersecurity threats

Model	R2L Recall	U2R Recall	Probe Recall	DoS Recall
Logistic Regression	0.0000	0.0000	0.3337	0.7274
Random Forest	0.0305	0.0150	0.7109	0.8034
XGBoost	0.0694	0.0350	0.7617	0.8162

While tree-based models demonstrate substantial improvements over Logistic Regression—particularly for Probe attacks (33% → 76%)—minority class detection remains critically inadequate. XGBoost detects only 191 of 2,754 R2L attacks (6.94%) and 7 of 200 U2R attacks (3.50%), representing unacceptable false negative rates for high-impact intrusion types.

7.4.2. Why Tree-Based Models Outperform Linear Models

The superior performance of Random Forest and XGBoost stems from their ability to capture the non-linear, hierarchical attack signatures present in the NSL-KDD dataset:

7.4.2.1. Complex Decision Boundaries

- Logistic Regression assumes linear separability—a single hyperplane to divide classes
- Attack patterns are inherently non-linear: DoS exhibits high error rates AND low same-service rates simultaneously
- Tree-based models create complex, multi-dimensional decision boundaries through recursive partitioning

7.4.2.2. Feature Interaction Capture

- Network intrusions are defined by feature combinations, not individual thresholds
- Example: S0 flag (connection attempt with no response) is strongly predictive only when combined with high connection counts and specific services
- Trees naturally model these interactions through hierarchical splits; linear models require manual interaction term engineering

7.4.2.3. Hierarchical Feature Structure

- NSL-KDD features are organized hierarchically (connection → service → host → host-service levels)
- Tree-based ensembles leverage this structure by splitting on different aggregation levels across trees
- Random Forest's feature randomization and XGBoost's gradient boosting both exploit these multi-scale patterns

7.4.3. The ROC-AUC vs Recall Disconnect

A critical finding emerges when comparing ROC-AUC scores with actual recall performance. The following table illustrates this disconnect for XGBoost, the best-performing model:

Attack Category	ROC-AUC	Recall	Gap Analysis
Probe	0.965	0.7617	Good alignment
DoS	0.961	0.8162	Good alignment
Normal	0.959	0.9226	Good alignment
U2R	0.928	0.0350	**Severe disconnect**
R2L	0.903	0.0694	**Severe disconnect**

All models demonstrate excellent discriminative ability (ROC-AUC > 0.90 for XGBoost), indicating they have learned to distinguish between attack types based on their probability distributions. However, this discrimination fails to translate into actionable predictions for minority classes. The ROC-AUC measures the model's ability to rank predictions correctly across all possible decision thresholds, while recall measures performance at the specific threshold chosen for classification.

For minority classes, severe class imbalance biases the optimal threshold toward majority class prediction. The model assigns higher probabilities to R2L and U2R attacks than to other classes—evidenced by strong AUC scores—but these probabilities remain too low relative to the imbalanced training distribution to trigger positive predictions. In essence, the model "knows" these are different attack types but lacks confidence to predict them given their extreme rarity in training data

7.4.4. Implications for Model Refinement

The baseline model evaluation reveals two critical insights for iterative refinement:

7.4.4.1. Class Imbalance Dominates Performance

- All three models demonstrate strong discriminative patterns (high ROC-AUC)
- Yet minority class recall remains catastrophically low (R2L: <7%, U2R: <4%)
- The barrier is not model capacity but class distribution in training data
- Solution: Synthetic Minority Over-sampling Technique (SMOTE) required to rebalance training data

7.4.4.2. XGBoost Provides Strongest Foundation

- Highest macro-averaged recall (0.5210)
- Best ROC-AUC across all classes (0.9432 macro)
- Superior minority class detection despite remaining inadequate
- Optimal candidate for SMOTE application and hyperparameter tuning

Without explicit class imbalance mitigation, no model architecture—regardless of sophistication—can achieve acceptable recall for rare but critical attack types. Iterative refinement phase will apply SMOTE to create synthetic R2L and U2R training examples, enabling models to learn decision boundaries calibrated for minority class detection rather than majority class optimization.

8. Iterative Refinement

The baseline models established in Section 7 demonstrated strong overall discrimination, with XGBoost achieving a ROC-AUC of 0.943. However, all three models exhibited severe performance failures on minority attack classes, with R2L recall below 7% and U2R recall below 4%. This section addresses these class imbalance issues through four progressively sophisticated approaches: synthetic oversampling (SMOTE), cost-sensitive learning, custom domain-motivated weighting strategies to be finally followed by hyperparameter tuning.

8.1. Handling Class Imbalance with SMOTE

8.1.1. Rationale

The training set exhibits extreme class imbalance: Normal traffic comprises 53.46% of records and DoS attacks 36.46%, while R2L represents just 0.79% (995 samples) and U2R only 0.04% (52 samples). Models trained on this distribution naturally optimise for the majority classes, as correctly classifying Normal and DoS records maximises overall accuracy while misclassifying all U2R records has negligible impact on the loss function.

SMOTE (Synthetic Minority Over-sampling Technique) addresses this by generating synthetic samples for minority classes. For each minority sample, SMOTE identifies its k nearest neighbours in feature space and creates new samples by interpolating between the original point and a randomly selected neighbour. This expands the minority class representation without simply duplicating existing records.

8.1.2. Strategy

Rather than equalising all classes to the majority count of 67,343, a conservative oversampling strategy was adopted. R2L was oversampled from 995 to 5,000 samples and U2R from 52 to 2,000 samples. DoS, Normal, and Probe were left unchanged. The rationale for this conservative approach was that generating 67,000+ synthetic samples from only 52 real U2R points would flood the training set with low-diversity synthetic data, likely degrading generalisation rather than improving it.

Class	Before SMOTE	After SMOTE	Change
DoS	45,927 (36.46%)	45,927 (34.81%)	No change
Normal	67,343 (53.46%)	67,343 (51.05%)	No change
Probe	11,656 (9.25%)	11,656 (8.84%)	No change
R2L	995 (0.79%)	5,000 (3.79%)	+4,005 synthetic
U2R	52 (0.04%)	2,000 (1.52%)	+1,948 synthetic

SMOTE was applied with k_neighbors=5 (the default), which is feasible even for U2R with 52 samples. The total training set increased from 125,973 to 131,926 samples (+5,953 synthetic). Critically, SMOTE was applied only to training data; the test set remained unchanged to ensure evaluation reflects real-world class distributions.

8.1.3. Results

SMOTE produced mixed results across the three model types. Logistic Regression performance actually degraded, with macro average recall falling from 0.401 to 0.334. The synthetic samples introduced noise that the linear model could not navigate, and Probe recall collapsed from 0.334 to effectively zero. Random Forest showed marginal improvement, with U2R recall increasing from 0.015 to 0.065 but R2L recall slightly declining. XGBoost benefited most from SMOTE, with R2L recall improving from 0.069 to 0.112 and U2R from 0.035 to 0.090, alongside improved macro F1 (0.526 to 0.579).

Model	R2L	U2R	Probe	DoS
Logistic Regression	0.000	0.000	0.334	0.727
LR + SMOTE	0.000	0.000	0.002	0.759
Random Forest	0.031	0.015	0.711	0.803
RF + SMOTE	0.026	0.065	0.714	0.825
XGBoost	0.069	0.035	0.762	0.816
XGB + SMOTE	0.112	0.090	0.821	0.817

While the XGBoost improvement is real, the results are ultimately disappointing. With only 52 U2R training samples, SMOTE interpolates within a very small region of feature space, producing synthetic points that lack the diversity needed to generalise to novel attack subtypes present in the NSL-KDD test set. This fundamental limitation motivated exploration of cost-sensitive learning approaches.

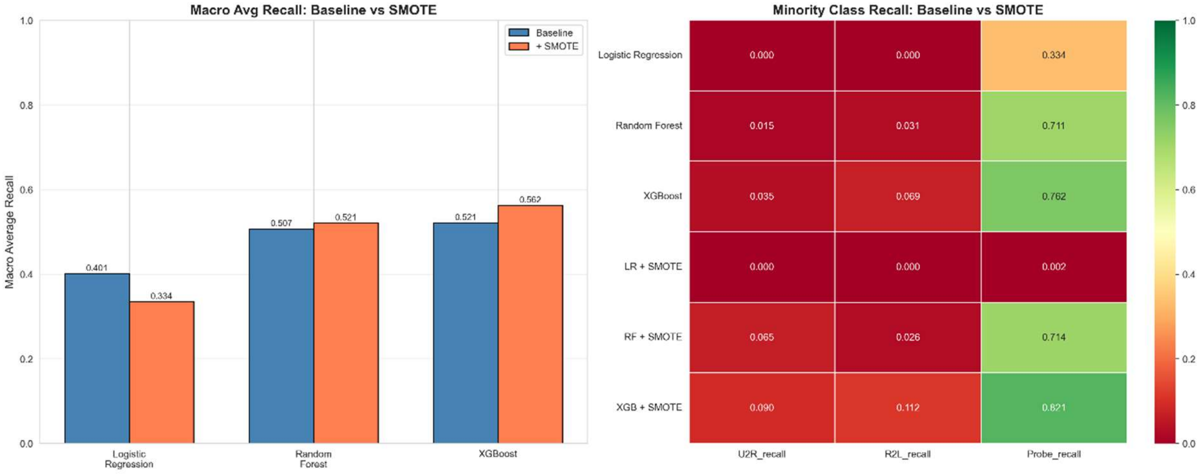


Figure 9: Macro Average Recall and Minority Class Recall Heatmap – Baseline vs SMOTE

8.2. Cost-Sensitive Learning

8.2.1. Approach

Where SMOTE addresses class imbalance by creating more training rows, cost-sensitive learning takes a fundamentally different approach: it modifies the loss function so that misclassifying minority samples incurs a much larger penalty during training. Rather than manufacturing synthetic data from limited examples, the model is told that getting a U2R prediction wrong costs far more than getting a Normal prediction wrong. This causes larger gradient updates when minority samples are misclassified, pushing the model to draw decision boundaries more conservatively around rare classes.

Using class_weight='balanced', scikit-learn automatically computes weights inversely proportional to class frequency using the formula: weight = n_samples / (n_classes × n_samples_per_class). For this dataset, this produces the weights shown in the table below:

Class	Training Samples	Weight	Effect
DoS	45,927	0.55x	Suppressed (abundant)
Normal	67,343	0.37x	Suppressed (most abundant)
Probe	11,656	2.16x	Slight boost
R2L	995	25.32x	Strong boost
U2R	52	484.51x	Extreme boost

Each U2R misclassification therefore costs approximately 484 times more than a Normal misclassification during training. This approach has a theoretical advantage over SMOTE for extreme imbalance: it works with the original 52 real U2R data points but makes each one enormously influential during optimisation, rather than relying on synthetic data quality.

8.2.1.1. Approach A: Balanced Weights Alone

All three models were retrained with class_weight='balanced' on the original (non-SMOTE) training data. For XGBoost, which does not natively support class_weight, equivalent per-sample weights were computed using sklearn's compute_sample_weight function.

The results revealed markedly different responses across model types. Logistic Regression showed a dramatic improvement in U2R recall, jumping from 0.000 to 0.455, and R2L recall from 0.000 to 0.260. However, this came at severe cost to overall accuracy (0.683 to 0.508), as the model aggressively reclassified many Normal samples as minority classes. The U2R F1 score of just 0.023 confirms that precision was near zero – the model detected many U2R attacks but generated an overwhelming number of false positives.

Random Forest responded poorly to balanced weights, with U2R recall actually dropping to 0.000 and R2L to 0.001. The 484x weight for U2R from only 52 samples distorted the tree splitting criteria so severely that the forest lost its overall discriminative structure. XGBoost handled the weights most effectively, achieving R2L recall of 0.242 (up from 0.069 baseline) and U2R recall of 0.095 (up from 0.035), while maintaining the highest ROC-AUC of any configuration tested at 0.952.

8.2.1.2. Approach B: SMOTE Combined with Balanced Weights

This approach applies both techniques: SMOTE first increases minority sample counts, then balanced weights further penalise misclassification. An important nuance is that balanced weights are computed from the SMOTE-rebalanced distribution, so the weights are less extreme than Approach A (since SMOTE has already partially balanced the classes).

XGBoost + SMOTE + Weighted achieved the best macro average recall (0.580) and macro F1 (0.613) of any configuration, with R2L recall of 0.225 and U2R recall of 0.120. While the combined approach outperformed either technique alone on aggregate metrics, it did not produce a clear breakthrough on minority class detection. Random Forest again performed poorly with combined techniques, confirming that this algorithm is less suited to extreme reweighting.

8.2.1.3. Approach C: Custom Domain-Motivated Weights

The automatically computed balanced weights treat all misclassification types as equally undesirable, differing only by class frequency. In cybersecurity, however, the severity of different attack types varies significantly. A U2R attack (root access escalation) represents a catastrophic breach where an attacker gains full system control, can exfiltrate data, install persistent backdoors, and pivot to other systems. An R2L attack (remote to local access) represents a serious breach but with more limited scope. A DoS attack, while disruptive to system availability, is recoverable once the attack ceases – no data is exfiltrated and no persistent access is gained.

Custom weights were designed to reflect these severity differences while also accounting for sample abundance. DoS and Normal classes were further suppressed not because they are unimportant, but because the model already has abundant training examples for these classes and does not need additional weighting to learn their patterns. R2L was given 50x weight (approximately double the balanced weight of 25.32x) and U2R was given 1000x (approximately double the balanced weight of 484.51x).

Class	Balanced Weight	Custom Weight	Rationale
DoS	0.55x	0.30x	Recoverable attack; abundant training data
Normal	0.37x	0.30x	Lowest risk; most abundant class
Probe	2.16x	2.00x	Keep stable
R2L	25.32x	50.0x	Serious breach, 2x balanced
U2R	484.51x	1000.0x	Catastrophic breach, 2x balanced

Custom weights produced the most extreme results of any strategy. Logistic Regression achieved a remarkable U2R recall of 0.890 – the highest of any model configuration – but at catastrophic cost to overall performance: accuracy collapsed to 18.8% and macro F1 fell to 0.227. The model essentially became a U2R/R2L detector at the expense of all other classes, with Probe recall dropping to 0.207 and DoS recall to 0.238. This demonstrates that a linear model with extreme weights will aggressively reclassify the majority of samples as minority classes.

Random Forest again responded poorly to aggressive weighting, with U2R recall of just 0.010 and R2L recall of 0.003 – worse than the baseline. XGBoost + Custom Weights achieved a more balanced result: U2R recall of 0.140 and R2L recall of 0.168, with overall accuracy of 77.9% and ROC-AUC of 0.951. XGBoost + SMOTE + Custom pushed U2R recall to 0.245 (the second-highest after LR + Custom) with R2L recall of 0.155, though at slightly lower overall F1 than the balanced weight equivalents.

The custom weights results reveal a key insight: more aggressive weighting does not uniformly improve minority detection. For XGBoost, the balanced weights (Approach A) actually achieved better R2L recall (0.242) than custom weights (0.168), suggesting that doubling the weight beyond balanced overcorrected for R2L while providing some benefit for U2R. The optimal weight configuration is model-dependent and class-dependent, with no single weighting strategy dominating across all metrics.

8.3. Summary of Imbalance Strategies

The below table presents the full comparison across all 15 model-strategy combinations tested. This comprehensive view reveals both the progress made and the persistent challenges in minority class detection.

Model	Macro Recall	Macro F1	AUC	Accuracy	R2L Recall	U2R Recall
Logistic Regression	0.401	0.397	0.790	0.683	0.000	0.000
Random Forest	0.507	0.508	0.891	0.765	0.031	0.015
XGBoost	0.521	0.526	0.943	0.758	0.069	0.035
LR + SMOTE	0.334	0.293	0.789	0.644	0.000	0.000
RF + SMOTE	0.521	0.529	0.881	0.773	0.026	0.065
XGB + SMOTE	0.562	0.579	0.944	0.791	0.112	0.090
LR + Weighted	0.449	0.429	0.775	0.508	0.260	0.455
RF + Weighted	0.472	0.470	0.885	0.741	0.001	0.000
XGB + Weighted	0.575	0.609	0.952	0.794	0.242	0.095
RF + SMOTE + Weighted	0.483	0.491	0.891	0.742	0.006	0.050
XGB + SMOTE + Weighted	0.580	0.613	0.950	0.792	0.225	0.120
LR + Custom Weights	0.343	0.227	0.669	0.188	0.270	0.890
RF + Custom Weights	0.477	0.477	0.880	0.749	0.003	0.010
XGB + Custom Weights	0.568	0.596	0.951	0.779	0.168	0.140
XGB + SMOTE + Custom	0.579	0.599	0.949	0.771	0.155	0.245

The results demonstrate that no single technique solves extreme class imbalance. Each approach offers trade-offs rather than clear solutions, and every improvement in minority class recall comes at some cost to overall accuracy or majority class precision.

Different models respond differently to rebalancing techniques. XGBoost consistently produced the best results across all strategies, handling both SMOTE and class weights gracefully through its boosting mechanism. Random Forest performed well with SMOTE (more data points) but poorly with extreme class weights, which distorted its tree-splitting criteria. Logistic Regression showed the most dramatic response to class weights, achieving the highest U2R recall of any model (0.890 with custom weights) but at the cost of near-random overall accuracy.

The trade-off between minority detection and false alarms is domain-relevant. In cybersecurity, missing an intrusion (false negative) is generally more costly than investigating a false alarm (false positive). This makes recall-optimised configurations defensible even when overall accuracy declines. A security operations centre would rather investigate false U2R alerts than miss a single real root access breach.

The persistent difficulty with U2R and R2L reflects a dataset limitation. The NSL-KDD test set deliberately includes novel attack subtypes absent from the training set. No amount of resampling or reweighting of known attack patterns can fully address the challenge of detecting previously unseen attack variants. This is a well-documented characteristic of the NSL-KDD benchmark, with published research typically reporting U2R recall in the 0.10–0.30 range using standard machine learning approaches.

The choice of best model depends on the operational priority:

Priority	Best Model	Score
Best Overall Minority Detection	**XGB + SMOTE + Weighted**	macro_avg_recall = 0.580
Best Overall Balance	**XGB + SMOTE + Weighted**	macro_avg_f1 = 0.613
Best U2R Detection	**LR + Custom Weights**	U2R_recall = 0.890
Best R2L Detection	**LR + Custom Weights**	R2L_recall = 0.270
Best Discrimination	**XGB + Weighted**	ROC-AUC = 0.952
Best Raw Accuracy	**XGB + Weighted**	accuracy = 0.794

For a balanced cybersecurity deployment, XGB + Weighted and XGB + SMOTE + Weighted represent the strongest candidates, combining competitive overall performance with meaningful minority class detection. These models will be taken forward to hyperparameter tuning in the next section.

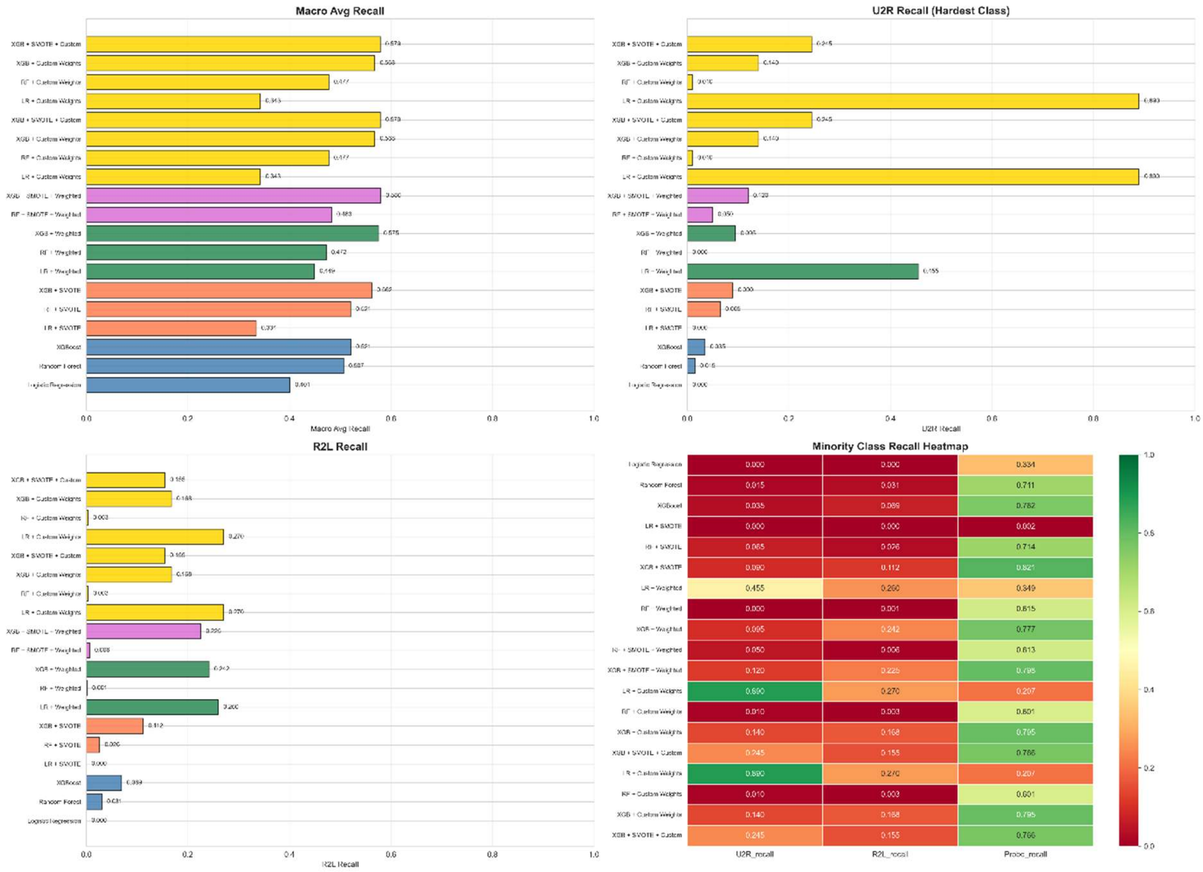


Figure 10: Full Comparison – Macro Average Recall, U2R Recall, R2L Recall, and Minority Class Heatmap.

8.4. Hyperparameter Tuning

The imbalance strategy comparison identified XGB + Weighted and XGB + SMOTE + Weighted as the two strongest candidates. Both used XGBoost with default hyperparameters (max_depth=6, n_estimators=100, learning_rate=0.1). This section explores whether optimising these hyperparameters can further improve performance, particularly on the minority attack classes where gains are most operationally valuable.

8.4.1. Methodology

RandomizedSearchCV was used to search over nine hyperparameters, sampling 100 random combinations from a space of over 1.4 million possibilities. Each combination was evaluated using 3-fold cross-validation, scored on macro-averaged F1 to balance recall and precision equally across all five classes. This yielded 300 model fits per candidate (600 total), completing in approximately 94 minutes.

RandomizedSearchCV was chosen over exhaustive GridSearchCV because research by Bergstra and Bengio (2012) demonstrated that random search finds near-optimal parameters more efficiently than grid search. Most hyperparameters have only a few dimensions that significantly affect performance; random search explores these important dimensions effectively rather than exhaustively testing every combination of parameters, with minimal impact.

Parameter	Values Tested
max_depth	3, 4, 5, 6, 7, 8, 10
n_estimators	100, 200, 300, 500
learning_rate	0.01, 0.05, 0.1, 0.2, 0.3
min_child_weight	1, 3, 5, 7, 10
subsample	0.6, 0.7, 0.8, 0.9, 1.0
colsample_bytree	0.6, 0.7, 0.8, 0.9, 1.0
gamma	0, 0.1, 0.2, 0.5, 1.0
reg_alpha	0, 0.01, 0.1, 1.0
reg_lambda	0.5, 1.0, 2.0, 5.0

8.4.2. Results

The two models found substantially different optimal configurations, reflecting their different training data characteristics.

Parameter	XGB + Weighted	XGB + SMOTE + Weighted
colsample_bytree	0.8	0.8
gamma	0.0	0.1
learning_rate	0.01	0.3
max_depth	6	10
min_child_weight	7	5
n_estimators	200	200
reg_alpha	0.1	0.01
reg_lambda	2.0	5.0
subsample	1.0	0.6

The most striking difference is in learning rate: XGB + Weighted selected a conservative 0.01 while XGB + SMOTE + Weighted preferred an aggressive 0.3. This makes intuitive sense. The weighted model uses extreme sample weights (up to 484x for U2R), so a slow learning rate prevents the model from overcorrecting to individual high-weight samples. The SMOTE model has more balanced training data (with synthetic minority samples), so it can afford faster learning without instability. Similarly, the SMOTE model selected max_depth=10 with subsample=0.6, indicating it benefits from deeper trees but requires subsampling to prevent overfitting to the synthetic data.

The table below presents the pre-tuning versus post-tuning comparison on the held-out test set.

Model	Macro Recall	Macro F1	AUC	Accuracy	R2L Recall	U2R Recall
XGB + Weighted	0.575	0.609	0.952	0.794	0.242	0.095
XGB + W (Tuned)	0.618	0.641	0.945	0.807	0.296	0.180
Change	**+0.043**	**+0.032**	**−0.007**	**+0.013**	**+0.054**	**+0.085**
XGB + SMOTE + W	0.580	0.613	0.950	0.792	0.226	0.120
XGB + SW (Tuned)	0.575	0.603	0.935	0.799	0.172	0.110
Change	**−0.005**	**−0.010**	**−0.015**	**+0.006**	**−0.054**	**−0.010**

XGB + Weighted (Tuned) showed substantial improvements across nearly all metrics. Macro average recall improved by 0.043, driven primarily by gains in minority class detection: R2L recall increased from 0.242 to 0.296 (+22%) and U2R recall nearly doubled from 0.095 to 0.180 (+89%). Probe recall also improved markedly from 0.777 to 0.861. Overall accuracy increased from 0.794 to 0.807, demonstrating that the minority class gains did not come at the expense of majority class performance. The only metric that declined was ROC-AUC, dropping slightly from 0.952 to 0.945. This is expected: the tuner optimised for macro F1 rather than AUC, so the decision boundaries shifted to favour actual classification accuracy over ranking discrimination.

XGB + SMOTE + Weighted (Tuned) showed the opposite pattern, with slight degradation across most metrics. Macro recall fell by 0.005, R2L recall dropped from 0.226 to 0.172, and AUC declined from 0.950 to 0.935. The cross-validation score during tuning was 0.999, compared to 0.965 for the weighted model. This near-perfect CV score is a strong indicator of overfitting: the SMOTE-generated synthetic samples are interpolations of real data, so cross-validation folds contain near-duplicate samples in both train and validation splits. The tuner found parameters that exploited this synthetic data structure rather than learning genuinely generalisable patterns, resulting in poorer test set performance.

8.4.3. Updated Best Model Summary

With tuning complete, the best model by metric table can be updated. XGB + Weighted (Tuned) now claims four of six categories, confirming it as the strongest overall model from this analysis.

Priority	Best Model	Score
Best Overall Minority Detection	XGB + Weighted (Tuned)	macro_recall = 0.618
Best Overall Balance	XGB + Weighted (Tuned)	macro_F1 = 0.641
Best U2R Detection	LR + Custom Weights	U2R_recall = 0.890

Best R2L Detection	XGB + Weighted (Tuned)	R2L_recall = 0.296
Best Discrimination	XGB + Weighted	ROC-AUC = 0.952
Best Raw Accuracy	XGB + Weighted (Tuned)	accuracy = 0.807

The tuning results reinforce the finding from Section 8.3 that XGBoost with balanced class weights represents the most effective approach for this dataset. Hyperparameter tuning provided meaningful gains on top of the imbalance handling strategy, particularly for minority class detection. The failure of tuning to improve the SMOTE + Weighted configuration further confirms that SMOTE's limitations with extremely small minority classes (52 U2R samples) represent a fundamental constraint rather than a tuning problem.

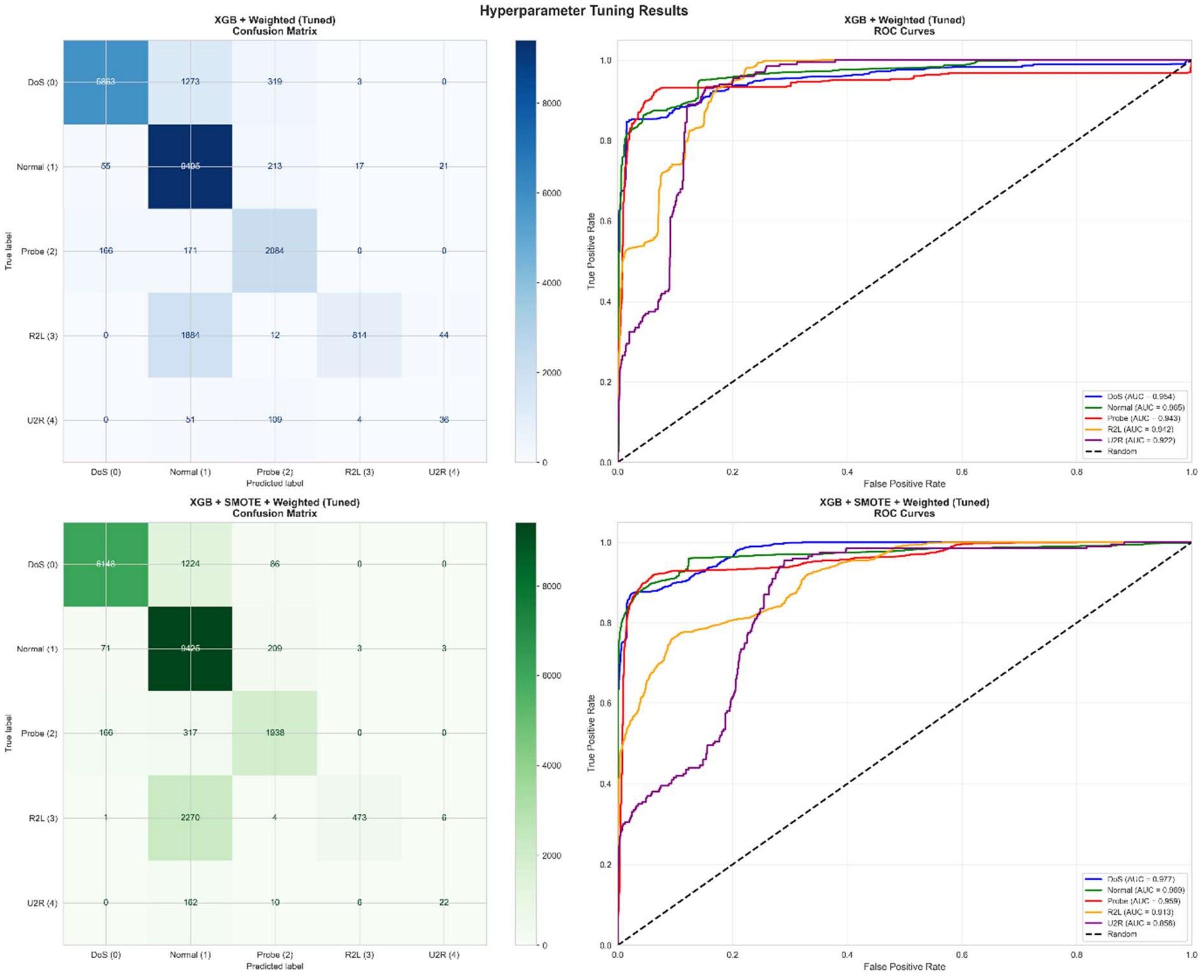


Figure 11: Hyperparameter Tuning Results — Confusion Matrices and ROC Curves for XGB + Weighted (Tuned) and XGB + SMOTE + Weighted (Tuned).

XGB + Weighted (Tuned) will be taken forward as the recommended model in the final model selection (Section 9).

9. Final Model Selection

The recommended model for network intrusion detection on the NSL-KDD dataset is XGBoost with balanced class weights and tuned hyperparameters (XGB + Weighted, Tuned). This model achieves the strongest overall balance between minority class detection and operational reliability, with a macro-averaged F1 of 0.641, macro-averaged recall of 0.618, and overall accuracy of 80.7%. It outperformed all 16 other model-strategy combinations tested on four of six evaluation criteria, including the two most operationally relevant: overall minority detection (macro recall) and overall balance between detection and false alarms (macro F1).

The choice prioritises prediction over interpretation. While Logistic Regression offers greater explainability through its coefficients, its severe performance failures on minority classes (R2L and U2R recall of zero at baseline) make it unsuitable for a cybersecurity application where missing even a single root access breach could have catastrophic consequences. XGBoost, as a gradient-boosted ensemble, sacrifices direct interpretability for substantially stronger predictive performance. This trade-off is appropriate for an intrusion detection system where the primary business value lies in correctly identifying attacks across all categories, and where feature importance analysis (discussed in Section 10.4) still provides meaningful insight into which network connection attributes are most predictive of malicious activity.

Metric	DoS	Normal	Probe	R2L	U2R	Macro Avg
Precision	0.964	0.736	0.761	0.971	0.356	0.758
Recall	0.786	0.969	0.861	0.296	0.180	0.618
F1-Score	0.866	0.836	0.808	0.453	0.239	0.641
Support	7,458	9,711	2,421	2,754	200	22,544

The model demonstrates strong and reliable performance on the three majority classes: Normal traffic is identified with 96.9% recall and 73.6% precision, DoS attacks with 78.6% recall and 96.4% precision, and Probe attacks with 86.1% recall and 76.1% precision. The precision-recall balance differs across these classes in an operationally sensible way: the model is conservative with DoS predictions (high precision, fewer false alarms) while being more aggressive with Normal classification (high recall, ensuring legitimate traffic passes through).

Minority class performance, while substantially improved over baseline, remains the primary limitation. R2L recall of 29.6% means approximately 70% of remote-to-local attacks are missed, though those that are detected have very high confidence (97.1% precision). U2R recall of 18.0% with 35.6% precision reflects the fundamental challenge of detecting novel root access attacks from only 52 training samples. These limitations are discussed further in Section 11.

The following table summarises the performance progression through the modelling pipeline, demonstrating that each refinement stage contributed meaningful improvement.

Stage	Macro Recall	Macro F1	AUC	Accuracy	R2L Recall	U2R Recall
XGBoost Baseline	0.521	0.526	0.943	0.758	0.069	0.035
+ Balanced Weights	0.575	0.609	0.952	0.794	0.242	0.095
+ Hyperparameter Tuning	0.618	0.641	0.945	0.807	0.296	0.180
Cumulative Improvement	**+0.097**	**+0.115**	**+0.002**	**+0.049**	**+0.227**	**+0.145**

The cumulative effect of balanced class weights and hyperparameter tuning increased R2L recall by 0.227 (from 6.9% to 29.6%) and U2R recall by 0.145 (from 3.5% to 18.0%), while also improving overall accuracy from 75.8% to 80.7%. The ROC-AUC remained essentially stable at 0.945, confirming that the model's underlying discrimination ability was preserved throughout the refinement process.

10. Key Findings and Insights

This analysis of the NSL-KDD dataset for network intrusion detection yielded several findings relevant both to the specific classification task and to the broader challenge of applying machine learning to cybersecurity data.

10.1. Class Imbalance Dominates Model Performance

The single most important factor affecting model quality was not the choice of model but the handling of class imbalance. The training set contains 67,343 Normal samples and 45,927 DoS samples but only 995 R2L and 52 U2R samples. Every baseline model, regardless of sophistication, effectively ignored the minority classes because the loss function provided no incentive to learn their patterns. Addressing this through cost-sensitive learning (balanced class weights) produced larger performance gains than switching between fundamentally different models. This finding underscores that in severely imbalanced domains, data strategy matters more than model selection.

10.2. XGBoost Is Robust to Rebalancing Strategies

Of the three algorithms tested, XGBoost consistently produced the best results across all imbalance handling strategies: SMOTE, balanced weights, custom weights, and combined approaches. Its gradient boosting mechanism handles extreme sample weights gracefully, progressively correcting errors on difficult (minority) samples across successive trees. Random Forest, by contrast, performed well with SMOTE (additional data points) but poorly with aggressive class weights, which distorted its tree-splitting criteria. Logistic Regression showed the most dramatic response to reweighting, achieving the highest U2R recall of any model (89.0% with custom weights) but at the cost of near-random overall accuracy (18.8%). These contrasting behaviours highlight that the interaction between model architecture and rebalancing technique is non-trivial and must be evaluated empirically.

10.3. SMOTE Has Fundamental Limitations with Extreme Minority Classes

SMOTE improved XGBoost's minority class recall modestly (U2R from 3.5% to 9.0%) but fell short of the gains achieved through cost-sensitive learning alone (U2R to 9.5% with balanced weights). With only 52 real U2R samples, SMOTE generates synthetic points by interpolating within a very small region of feature space, producing data that lacks the diversity needed to generalise to novel attack subtypes. The hyperparameter tuning results reinforced this finding: the SMOTE-based model achieved a near-perfect cross-validation score of 0.999 (indicating overfitting to synthetic data) but degraded on the held-out test set. For datasets with fewer than approximately 100 minority samples, cost-sensitive learning appears to be the more effective approach.

10.4. Feature Importance Reveals Attack Detection Mechanisms

XGBoost's built-in feature importance scores (based on the total gain contributed by each feature across all trees) reveal which network connection attributes the model relies on most heavily to distinguish between traffic classes. The top five features alone account for 45.7% of total model importance, and the top ten account for 64.5%, indicating that the model's decision-making is concentrated on a relatively compact set of informative features from the 47 available.

Rank	Feature	Importance	Interpretation
1	flag_S0	0.190	Connection attempted, no response – SYN flood / port scan signature
2	root_shell	0.107	Root access obtained – primary U2R attack indicator
3	service_encoded	0.058	Target-encoded service type – certain services attract specific attacks
4	srv_diff_host_rate	0.058	Connections to different hosts on same service – scanning behaviour
5	dst_host_serror_rate	0.044	SYN error rate at destination – DoS/Probe signature
6	protocol_type_tcp	0.043	TCP protocol flag – most attacks use TCP connections
7	duration	0.041	Connection duration – DoS attacks are typically very short
8	count	0.039	Connections to same host in window – burst activity indicator
9	src_bytes	0.033	Bytes sent by source – data exfiltration / payload size
10	dst_bytes	0.033	Bytes sent by destination – response volume patterns

The most important feature by a substantial margin is `flag_S0` (19.0%), which indicates a connection where a SYN packet was sent but no response was received. This is the characteristic signature of both SYN flood DoS attacks and port scanning (Probe) activity, making it the single most discriminating feature in the dataset. The second most important feature, `root_shell` (10.7%), is a binary indicator of whether root access was obtained during the connection. This feature is the model's primary mechanism for identifying U2R attacks — the most dangerous and rarest attack category. Its high ranking despite U2R comprising only 0.04% of training data demonstrates that the balanced class weights successfully elevated this feature's influence during training.

The remaining top features fall into three interpretable categories. Connection pattern features (`srv_diff_host_rate, dst_host_serror_rate, count, dst_host_srv_count`) capture the statistical signatures of scanning and flooding behaviour, which are characteristic of Probe and DoS attacks respectively. Content features (`service_encoded, protocol_type_tcp, duration, src_bytes, dst_bytes`) reflect what data is being transferred and how, helping distinguish between legitimate traffic and attacks that exploit specific services or transfer anomalous volumes of data. Authentication features (`num_failed_logins, logged_in, is_guest_login`) are particularly relevant for R2L and U2R detection, where attackers attempt to gain progressively higher levels of access through repeated authentication attempts or credential exploitation.

These feature importance results corroborate the patterns identified during exploratory data analysis in Section 6. The EDA revealed that DoS attacks produce extreme values in connection rate features and that U2R attacks are strongly associated with authenticated sessions — both patterns are reflected in the model's learned feature rankings. Critically, features that appeared near-constant in overall distribution (such as `root_shell`, which is zero for the vast majority of connections) proved highly important for rare class detection, validating the decision made during feature engineering to retain these features rather than removing them as uninformative.

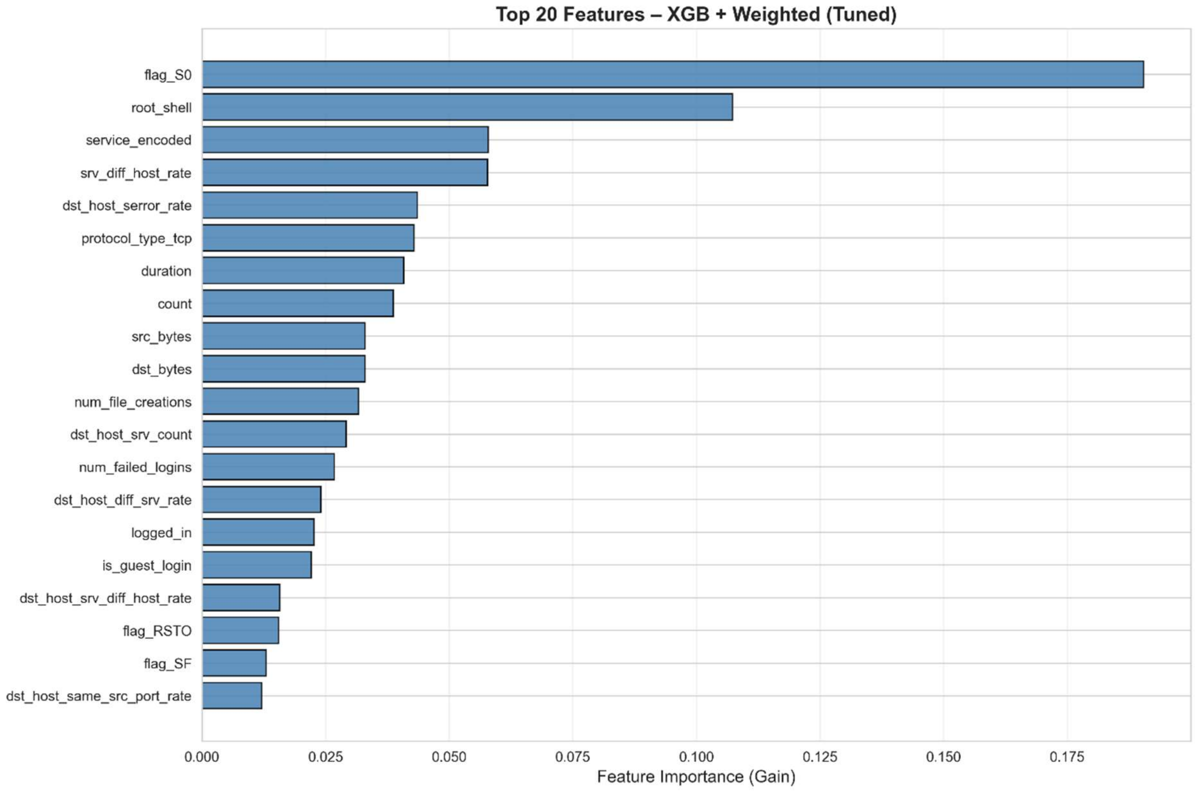


Figure 12: Top 20 Feature Importances – XGB + Weighted (Tuned).

10.5. The Recall-Precision Trade-Off Is Domain-Dependent

Every improvement in minority class recall came at some cost to precision or overall accuracy. The appropriate balance depends on the operational context. In a security operations centre, investigating a false U2R alert costs analyst time (perhaps 30 minutes), while missing a real root access breach could cost millions in data exfiltration, regulatory penalties, and reputational damage. This asymmetry makes recall-optimised configurations defensible even when they generate additional false positives. The recommended model achieves U2R precision of 35.6%, meaning roughly two out of three U2R alerts are false alarms. In a real deployment this would translate to manageable alert volumes given the low base rate of U2R traffic, and each alert would still warrant investigation given the catastrophic potential of a true positive.

11. Next Steps and Limitations

11.1. Dataset Limitations

The NSL-KDD dataset, while a widely used benchmark, has well-documented limitations that constrain the conclusions of this analysis. The test set deliberately includes novel attack subtypes absent from the training data, meaning no supervised model trained exclusively on known patterns can fully detect these unseen variants. This design choice reflects a real-world challenge (attackers constantly develop new techniques) but also means that the reported minority class recall figures represent a lower bound on what could be achieved with more representative training data. Additionally, the dataset originates from 1999 network traffic and does not reflect modern network protocols, encryption patterns, or contemporary attack vectors. Results should therefore be interpreted as demonstrating methodology rather than production-ready performance.

11.2. Model Limitations

The recommended model misses approximately 70% of R2L attacks and 82% of U2R attacks. While this is consistent with published benchmarks for standard machine learning on NSL-KDD, it would be insufficient for a standalone production intrusion detection system. The model also sacrifices direct interpretability by using XGBoost rather than a more transparent algorithm. While feature importance scores provide some insight, they do not offer the clear coefficient-based explanations that a logistic regression model would, which could be a limitation in regulated environments where model decisions must be fully explainable.

11.3. Recommended Next Steps

Several avenues could improve upon the results achieved in this analysis. First, incorporating anomaly detection alongside the supervised classifier could address the novel attack problem. An unsupervised model trained to recognise normal traffic patterns could flag statistical outliers regardless of whether they match known attack signatures, providing a complementary detection layer for zero-day attacks that the supervised model cannot learn from labelled examples alone.

Second, more sophisticated feature engineering could improve minority class detection. Network traffic data contains temporal patterns (sequences of connections, session-level behaviour) that flat tabular features cannot fully capture. Recurrent neural networks or transformer-based architectures applied to connection sequences might detect the subtle multi-step patterns characteristic of R2L and U2R attacks more effectively than tree-based models operating on individual connection records.

Third, the analysis could benefit from more advanced resampling techniques. Approaches such as ADASYN (Adaptive Synthetic Sampling), which generates more synthetic samples in regions of feature space where the classifier struggles, or borderline-SMOTE, which focuses synthetic generation near decision boundaries, may produce higher-quality synthetic minority samples than standard SMOTE. Alternatively, cost-sensitive deep learning approaches could learn non-linear representations of minority classes without relying on synthetic data at all.

Finally, applying this methodology to a modern intrusion detection dataset such as CICIDS2017 or CSE-CIC-IDS2018 would test whether the findings generalise to contemporary network traffic. These datasets contain more recent attack types, larger sample sizes, and richer feature sets that could address several of the limitations identified in this analysis.

12. References

NSL-KDD Dataset: https://www.unb.ca/cic/datasets/nsl.html

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. IEEE Symposium on Computational Intelligence for Security and Defense Applications.

Bergstra, J. and Bengio, Y. (2012) 'Random Search for Hyper-Parameter Optimization', *Journal of Machine Learning Research*, 13(Feb), pp. 281–305. Available at: https://jmlr.org/papers/v13/bergstra12a.html