Feature Descriptions & Preprocessing

---

1. Feature Overview

1.1 Time & Size Features

- `flow_time`
  - *Definition*: Total elapsed time from the first to the last packet in a flow (e.g. seconds or milliseconds).
  - *Usage*: Differentiates short (“chatty”) flows from long bulk transfers or anomalies.
- `header_size`
  - *Definition*: Total bytes of IP/TCP headers exchanged.
  - *Usage*: Higher header overhead may indicate many small packets or certain protocol behaviors.
- `packet_duration`
  - *Definition*: Average (or total) duration per packet within the flow.
  - *Usage*: Distinguishes interactive traffic (short durations) from bulk transfers (long durations).

1.2 Throughput / Rate Features

- `overall_rate`
  - *Definition*: Total bytes divided by `flow_time` (bytes/sec).
  - *Usage*: Captures end-to-end throughput of the flow.
- `src_rate` / `dst_rate`
  - *Definition*: Throughput in source→destination and destination→source directions.
  - *Usage*: Asymmetric rates often flag client/server roles or scanning.

1.3 Packet-Count & Flag-Count Features

- **Packet counts**

- `fin_packets`, `urg_packets`, `rst_packets`
  - *Definition*: Number of packets carrying the FIN, URG, or RST flags.
  - *Usage*:
    - `FIN` ⇒ normal teardown
    - `RST` ⇒ abrupt reset (scans, errors)
    - `URG` ⇒ rare urgent data
- **TCP-flag counts**
  - `fin_flags`, `syn_flags`, `rst_flags`, `psh_flags`, `ack_flags`
  - *Definition*: Number of packets with each TCP control flag set.
  - *Usage*:
    - `SYN` ⇒ connection initiation
    - `ACK` ⇒ acknowledgments
    - `PSH` ⇒ push (application data)
    - Patterns of these can reveal scans or DoS behavior.

1.4 Statistical Features

- `max_value`
  - *Definition*: Maximum observed value in a per-packet metric (e.g. packet size).
- `value_covariance`
  - *Definition*: Covariance (or variance) of a per-packet metric over the flow.
  - *Usage*: Quantifies burstiness or variability in packet sizes/timings.

1.5 Protocol Indicators (One-Hot)

- `protocol_http`, `protocol_https`, `protocol_tcp`, `protocol_udp`, `protocol_icmp`
  - *Definition*: Binary flags (0/1) indicating which protocol the flow used.
  - *Usage*: Encodes protocol type directly; mutually-exclusive in most cases.

1.6 Target Label

- `label`
  - *Definition*: Ground-truth class (e.g. benign vs. malicious, or specific attack types).

- *Usage*: Supervised target—**never** apply feature transforms directly to this column.

---

2. Preprocessing Techniques

Technique	Numeric Features	Binary / One-Hot Columns	Target (`label`)
Duplicate Removal	✔ Drop duplicate rows	— (no duplicates in flags)	— (labels untouched)
Skewness Fixing	✔ (log, Box–Cox)	— (binary → no skew)	— (not applicable)
Outlier Handling	✔ (IQR capping, trimming)	—	—
Scaling	✔ (standard, min–max, robust)	◯ (optional—for distance-based models)	—
Missing-Value Check	✔ (none found, no imputation)	—	—
Feature Selection	✔ (variance threshold, PCA)	✔ (e.g. remove collinear flags)	—
Class Rebalancing	✔ via SMOTE (continuous)	✔ via SMOTENC (mixed data)	⚠ Only on training labels

- **Notes on SMOTE/SMOTENC**
  - **SMOTE**: only for continuous features.
  - **SMOTENC**: for mixed continuous + binary data—specify which columns are categorical to preserve one-hot integrity.
  - Always apply oversampling **after** train/test split, and only to the **training** set.

---

3. Suggested Pipeline

1. **Global Deduplication**
   - Drop all exact-duplicate rows from the DataFrame.
2. **Train/Test Split**
   - Split into train & test **before** any transforms to avoid data leakage.
3. **Training-Set Preprocessing**

4. 1. **Fix skewness** on numerical features (e.g. `flow_time`, `overall_rate`).
5. 2. **Handle outliers** via IQR capping or winsorization.
6. 3. **Scale** continuous features (e.g. StandardScaler or MinMaxScaler).
7. 4. **Balance** the target using SMOTE (or SMOTENC if mixed data).
8. **Test-Set Transformation**
    - Apply the same skew-transform and scaler fitted on the training set.

> **Tip:** If you use tree-based models (e.g. Random Forest), you may skip scaling but still benefit from skew-fixing and outlier capping.
