

AI Techniques for Networking

Supervised Learning for Traffic Classification Lecture 3-4

By Dr. Tarfa Yaseen Hamed

Mar. 2026

Outline

- ▶ Introduction to Traffic Classification
- ▶ The Supervised Learning Pipeline
- ▶ Feature Engineering (Flow-based vs. Packet-based)
- ▶ Algorithms for Traffic Classification
- ▶ Comparison of Supervised Classifiers
- ▶ Data Acquisition - The Foundation of the Pipeline
- ▶ Capture Tools (The Engineer's Toolkit)
- ▶ Modern Challenge: Traffic Generation
- ▶ Feature Extraction

Introduction to Traffic Classification

The "Why" and "How"

- **Definition:** The process of identifying the type of application or protocol (e.g., YouTube, BitTorrent, SSH) generating a specific stream of network traffic.
- **Motivation:**
 - **Quality of Service (QoS):** Prioritizing latency-sensitive traffic (VoIP) over bulk data (Downloads).
 - ▶ **Security:** Detecting malicious patterns or unauthorized tunneling.
 - ▶ **Traffic Engineering:** Capacity planning based on application trends.
- ▶ **Note:** Traditional methods like Port-based (easily bypassed) and **Deep Packet Inspection (DPI)** (limited by encryption) are being replaced by **Machine Learning (ML)**.

The Supervised Learning Pipeline

To classify traffic using supervised learning, we follow a structured workflow:

1. **Data Collection:** Capturing raw traffic (PCAP files) from network taps or port mirroring.
2. **Flow Bifurcation:** Grouping packets into **flows** (usually defined by the 5-parts: Source/Dest IP, Source/Dest Port, Protocol).
3. **Feature Extraction:** Converting raw packets into numerical vectors (e.g., packet inter-arrival times, flow duration).
4. **Labeling (Ground Truth):** Assigning known labels to flows (the hardest part in supervised learning).
5. **Training & Evaluation:** Feeding the labeled data into a classifier.

Feature Engineering (Flow-based vs. Packet-based)

- Features are the "signals" the AI uses to make decisions.

Feature Category	Examples	Pros/Cons
Statistical (Flow)	Mean packet size, Flow duration, Variance of Inter-Arrival Time (IAT).	Resilient to encryption; requires waiting for flow completion.
Header-based	TCP Window size, TTL values, Flag counts.	Fast extraction; vulnerable to header obfuscation.
Payload-based	First N bytes of the payload (if unencrypted).	High accuracy; fails completely on HTTPS/TLS.

Algorithms for Traffic Classification

- **Decision Trees (DT) / Random Forest (RF):**

- *Strength:* Excellent interpretability (you can see the "rules").
- *Performance:* Usually the baseline for traffic classification due to high speed and accuracy.

- **Support Vector Machines (SVM):**

- *Strength:* Effective in high-dimensional spaces.
- *Weakness:* Computationally expensive for high-speed core networks.

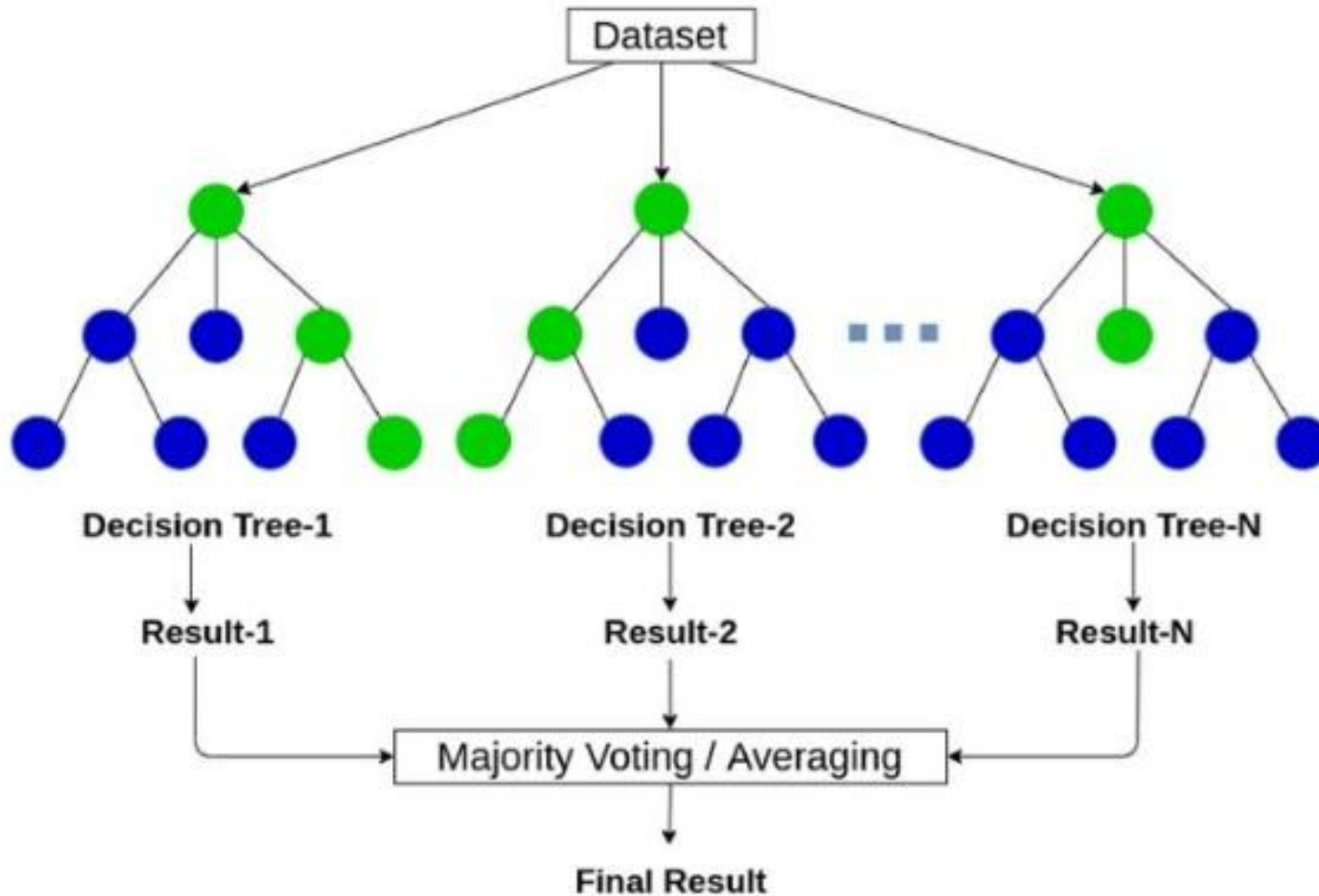
- **K-Nearest Neighbors (k-NN):**

- *Strength:* Simple; no formal "training" phase.
- *Weakness:* Very slow during inference (testing), as it must search the entire dataset.

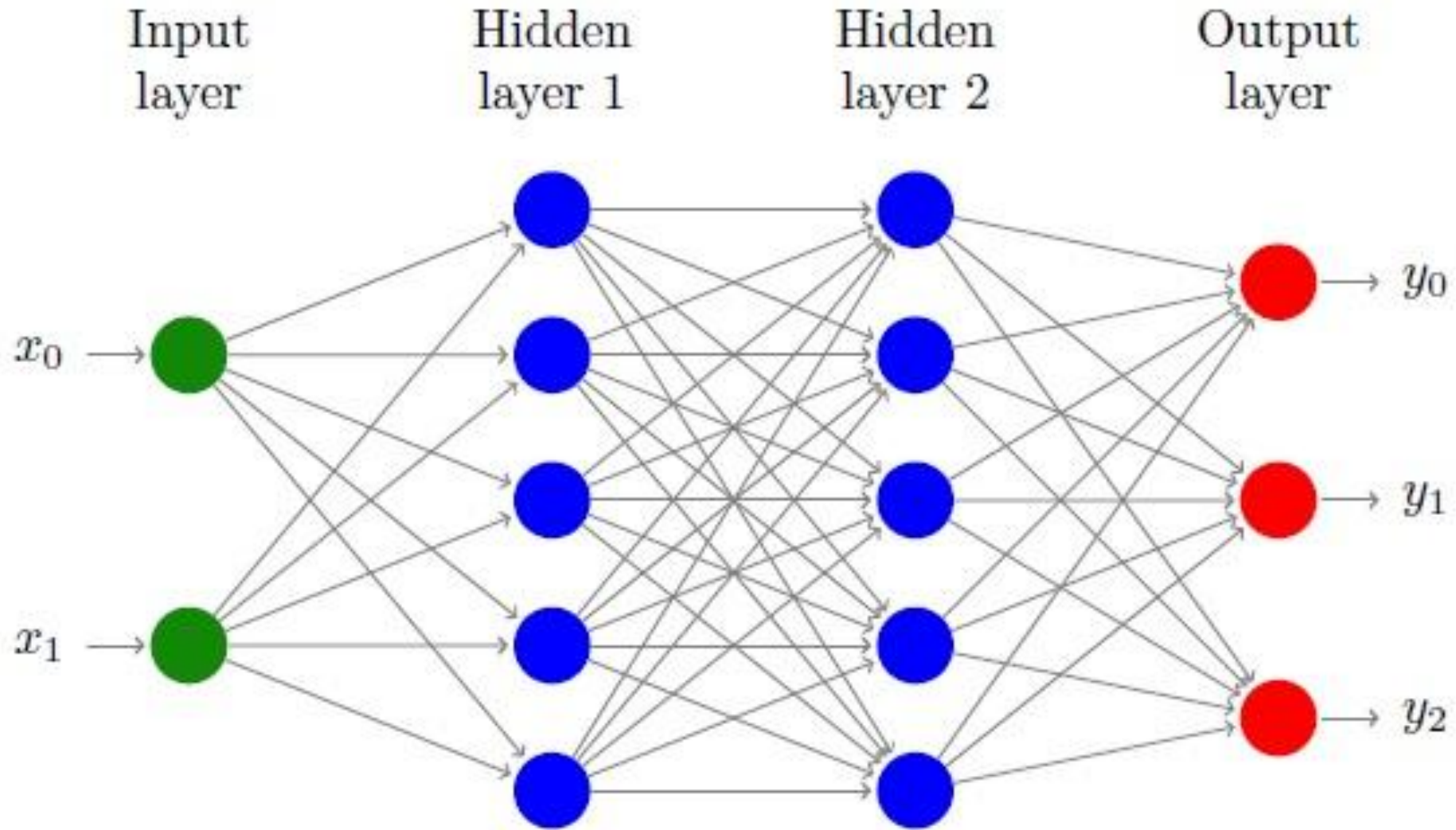
- **Neural Networks (MLP):**

- *Strength:* Can capture non-linear relationships in complex traffic patterns.

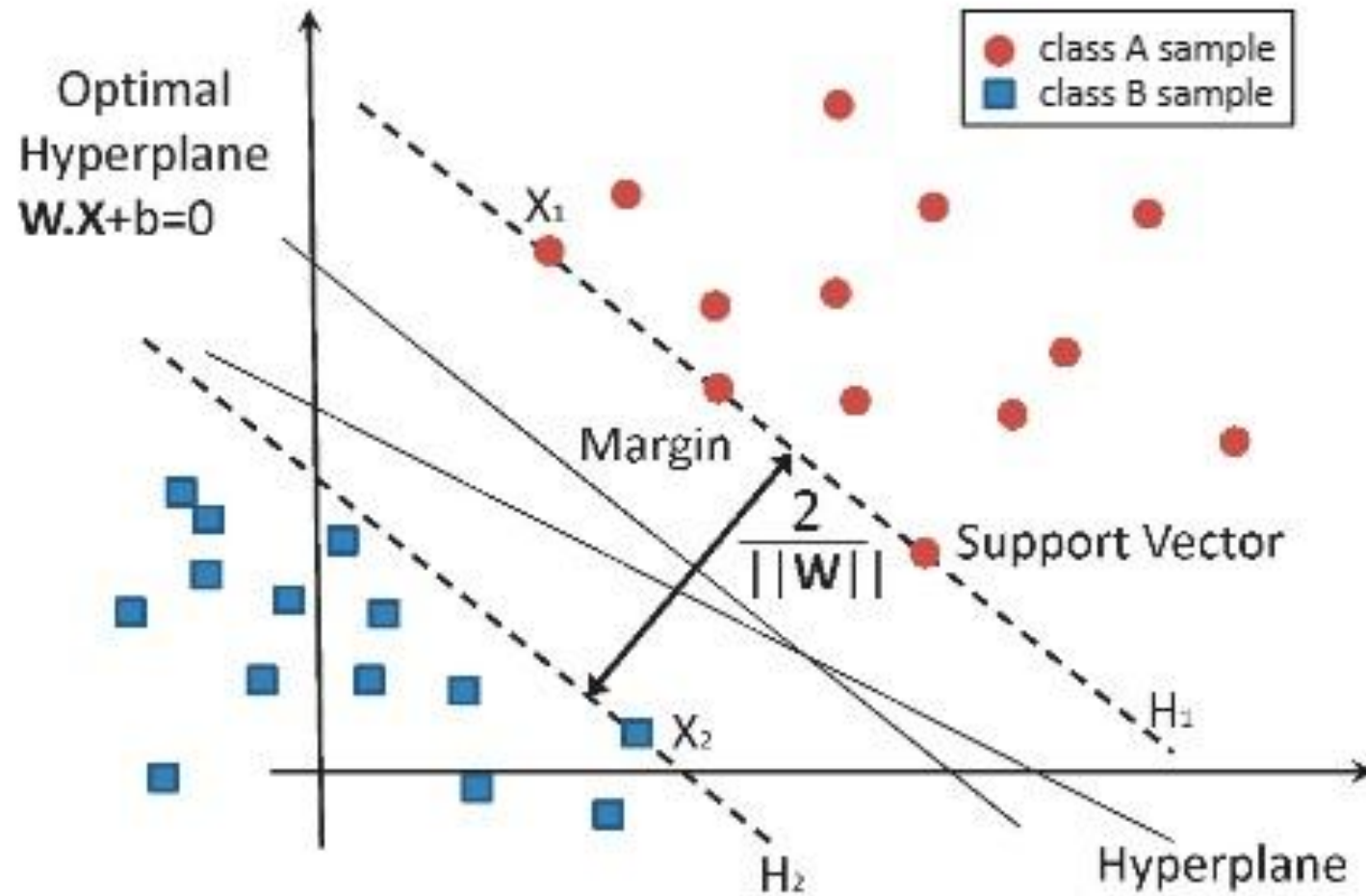
Decision Trees (DT) / Random Forest (RF):



Neural Networks (MLP):

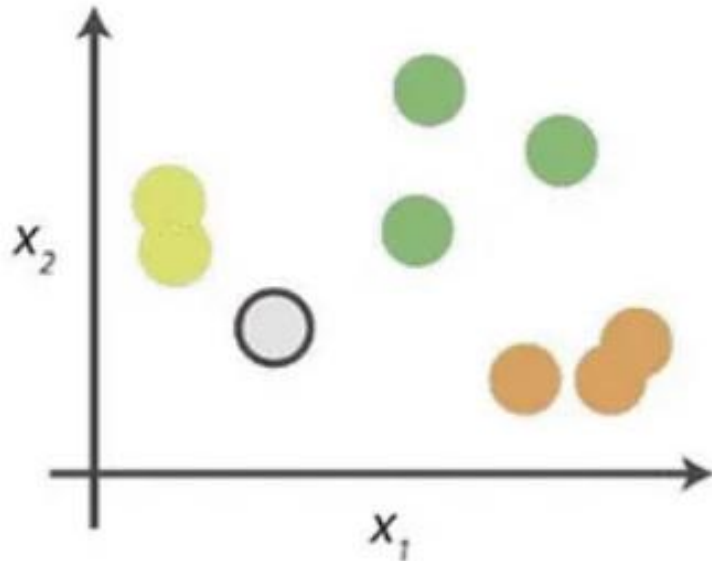


Support Vector Machines (SVM)



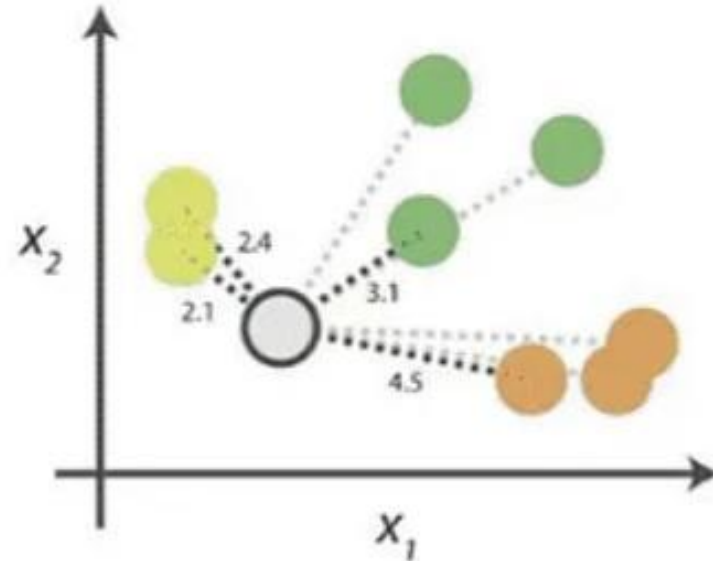
K-Nearest Neighbors (k-NN)

0. Look at the data



Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.

1. Calculate distances



Start by calculating the distances between the grey point and all other points.

K-Nearest Neighbors (k-NN) (cont.)

2. Find neighbours

Point Distance		
 ... 	2.1	→ 1st NN
 ... 	2.4	→ 2nd NN
 ... 	3.1	→ 3rd NN
 ... 	4.5	→ 4th NN

Next, find the nearest neighbours by ranking points by increasing distance. The nearest neighbours (NNs) of the grey point are the ones closest in dataspace.

3. Vote on labels

Class	# of votes	
	2	→ Class  wins the vote! Point  is therefore predicted to be of class  .
	1	
	1	

Vote on the predicted class labels based on the classes of the k nearest neighbours. Here, the labels were predicted based on the k=3 nearest neighbours.

Comparison of Supervised Classifiers

Based on common benchmark datasets (like ISCX or Moore):

Algorithm	Accuracy	Speed (Inference)	Memory Usage	Best For...
Random Forest	97-99%	Fast	Medium	General Application ID
Naive Bayes	70-85%	Very Fast	Low	Real-time edge filtering
<i>k-NN</i>	90-95%	Slow	High	Small datasets/Offline
XGBoost	98%+	Medium	Medium	State-of-the-art research

Data Acquisition - The Foundation of the Pipeline

- ▶ In supervised learning for networking, the quality of your model is strictly dependent on how you capture the "Ground Truth."

1. Physical Capture Methods

To train a model, we must intercept traffic without disrupting the network flow.

- **Port Mirroring (SPAN):** A software-based approach where a switch sends a copy of network packets seen on specific ports (or an entire VLAN) to a network monitoring port.
 - *Constraint:* Can drop packets during high CPU utilization on the switch.
- ▶ **Network TAPs (Test Access Points):** A hardware device inserted directly into a network cable. It "splits" the signal to a monitoring device.
 - *Advantage:* Provides an exact copy of all traffic (including physical layer errors) with zero impact on network performance.

Capture Tools

- **Tcpdump:** A command-line packet analyzer. It is the gold standard for high-performance, headless capture in Linux environments.
- **Wireshark / TShark:** Used for deep inspection and protocol analysis. TShark (the CLI version) is often used in scripts to automate the generation of PCAP (Packet Capture) files.
- **Argus / Zeek:** These tools don't just capture packets; they perform "Live Flow Extraction," converting raw packets into connection-level summaries in real-time.

Data Formats: PCAP vs. NetFlow

	PCAP (Raw Packets)	NetFlow / IPFIX (Flow Data)
Contents	Full headers + Full payload	Metadata only (5-parts, byte count)
Storage	Extremely high (Terabytes/hour)	Low (summarized)
AI Use Case	Deep Learning (CNN/RNN)	Traditional Supervised (RF/SVM)
Privacy	High risk (contains user data)	Lower risk (metadata only)

Modern Challenge: Traffic Generation

To get "Labels" for supervised learning, engineers often use **Controlled Traffic Generation**:

- Running specific applications (e.g., opening only Spotify) in a clean virtual machine.
- Labeling that specific MAC address or IP as the "Target Class."
- **Tools: `iperf3`** for throughput simulation, or **Selenium** for automated web traffic generation.

Feature Extraction (Statistical) - Turning Packets into Vectors

- ▶ In supervised learning, raw packets cannot be fed directly into most algorithms (like Random Forest or SVM). First, the "raw wire data" must be transformed into a **Feature Vector**: a fixed-length numerical representation of a network flow.

1. The Concept of "Flow Statistics"

- ▶ Instead of looking at the content of a single packet, statistical extraction looks at the **behavior** of a group of packets (a flow) over time.
- ▶ This is the most powerful method for classifying **encrypted traffic**, as the statistics remain visible even when the payload is hidden.

Feature Extraction (Statistical) - cont.

2. Core Statistical Features

Features can be categorized into three main buckets:

Feature Category	Examples	Importance for AI
Packet Lengths	Min, Max, Mean, Std Dev, Median	Distinguishes between "chatty" protocols (Telnet) and "bulk" transfers (FTP).
Byte Counts	Total bytes sent, Payload-to-Header ratio	Identifies data-heavy applications like Video Streaming or File Sharing.
Flow Dynamics	Total duration, Packets per second (PPS)	Helps separate automated bot traffic from human-driven web browsing.

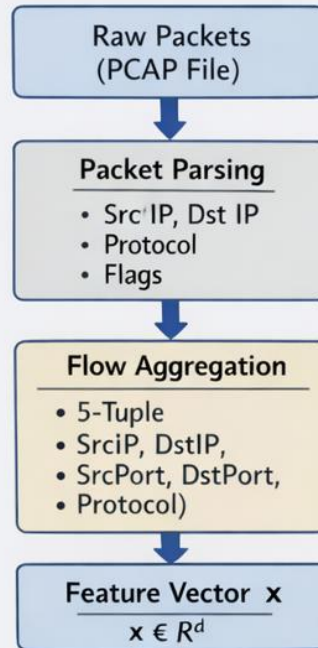
Feature Extraction (Statistical) - cont.

3. The Mathematical Representation

For a sequence of packet lengths $L = \{l_1, l_2, \dots, l_n\}$, we calculate:

- **Arithmetic Mean (μ):** $\mu = \frac{1}{n} \sum_{i=1}^n l_i$
- **Standard Deviation (σ):** $\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (l_i - \mu)^2}$

Packet-Level to Flow-Level

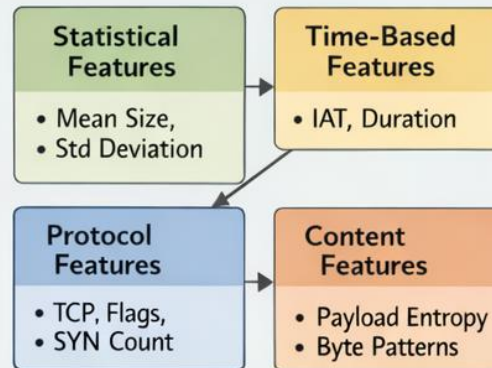


Statistical Features

Packet Lengths: [60, 60, 1500, 60, 60, 900, 60]

Mean = 386 bytes
Std Dev = 560
Variance = 313,600
Max = 1500
Min = 60

Feature Categories

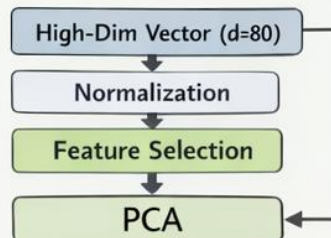


Feature Matrix

N Flows x d Features

	Dur	Bytes/s	IAT_mean	SYN_Count
Flow1	10	5000	0.002	2
Flow2	3	12000	0.0001	45
Flow3	50	200	0.5	0

Dimensionality Reduction



Mathematical Formulation

$$F = \{p_1, p_2, \dots, p_n\}$$

$$p_i = (t_i, l_i, f_i)$$

$$Duration = t_n - t_1$$

$$\mu_i = \frac{1}{n} \sum l_i$$

$$IAT_i = t_i - t_{i-1}$$

$$\mu_{IAT} = \frac{1}{n-1} \sum IAT_i$$

Entropy Calculation

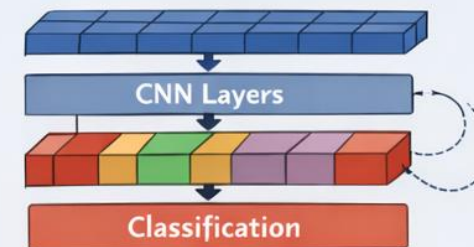
IP	Propability
A	0.7
B	0.2
C	0.1

$$H = - \sum_n p_i \log p_i$$

Low $H \rightarrow$ Focused Attack

High $H \rightarrow$ Distributed Traffic

Deep Feature Extraction



Why Statistical Features Win

- **Protocol Agnostic:** Works for TCP, UDP, and even non-IP protocols.
- **Encryption Resistance:** Whether the packet contains an encrypted YouTube video or an encrypted bank transaction, the *size* and *timing* of the packets will follow different statistical distributions.
- **Efficiency:** Summarizing 1,000 packets into 20 statistical features significantly reduces the computational load on the AI model.

Practical Example: Video vs. VoIP

- **Video Streaming (Netflix):** Large packet lengths (near MTU of 1500 bytes) with high variance.
- **VoIP (WhatsApp Call):** Small, consistent packet lengths (usually <200 bytes) with very low variance (steady stream of audio data).

The Netflix logo is displayed in red, bold, sans-serif capital letters on a black rectangular background.

Feature Extraction (Temporal) - The Rhythm of the Network

- ▶ While **Statistical Features** focus on "how much" data is sent, **Temporal Features** focus on "when" it is sent.
- ▶ This is often the most critical differentiator for AI models when identifying automated bot traffic versus human-induced traffic.

1. Understanding Inter-Arrival Time (IAT)

- ▶ The primary temporal feature is the **Inter-Arrival Time (IAT)**.
- ▶ This is the time difference between the arrival of two consecutive packets in a flow.
 - **Definition:** $IAT_n = T_n - T_{n-1}$, where T is the arrival timestamp.
 - **Forward IAT:** Time between packets sent from Source to Destination.
 - **Backward IAT:** Time between packets sent from Destination to Source.

2. Key Temporal Metrics

In a supervised learning dataset, it does not just use raw timestamps; it calculates the following "Time-Series" aggregates:

- **Flow Duration:** The total time from the first SYN (or first packet) to the last FIN (or timeout).
 - *Example:* A DNS query lasts milliseconds; a Netflix movie lasts hours.

Key Temporal Metrics(cont.)

- **Idle Time vs. Active Time:**

- **Active:** Periods where packets are being continuously exchanged.
- **Idle:** The "Think Time" (e.g., a user reading a webpage before clicking the next link).

- **Jitter:** The variation in the delay of received packets. High jitter is a signature of congested or low-quality wireless links.

The "Burst" (Near-Zero IAT)

- ▶ A **burst** occurs when a server tries to send as much data as possible in the shortest amount of time to fill your device's memory (the buffer).
 - **Behavior:** The server sends 50 or 100 packets back-to-back at the full speed of the link.
 - **The IAT (Inter-Arrival Time):** Because there is no intentional delay between these packets, the time difference between them is almost nothing—often measured in microseconds (μs).
 - **AI Interpretation:** "The network is currently in a high-throughput state; the application is fetching a chunk of data."

Why this helps Supervised Learning

- ▶ By looking at the "High vs. Near-Zero" IAT pattern, a supervised learning model (like a Decision Tree) can create a logic gate to identify the application:
 - **Near-zero IAT** = High-speed data delivery (filling the buffer).
 - **High IAT** = Idle time (waiting for the buffer to empty).
 - **The Contrast:** It is the difference between these two states that allows the AI to "fingerprint" streaming services, even when the content is encrypted.

3. Temporal "Fingerprinting" of Applications

Different applications have unique "heartbeats" that AI can easily recognize:

Application Type	Temporal Behavior	IAT Distribution
Video Streaming	Bursty	High IAT between bursts, near-zero IAT during a burst.
VoIP/Gaming	Periodic	Extremely consistent, low IAT (e.g., 20ms fixed intervals).
IoT Sensor	Heartbeat	High IAT (e.g., one packet every 60 seconds).
Malware (C2)	Beaconing	Fixed, repetitive IATs used to check for commands.

Challenges in Temporal Extraction

1. **Network Jitter:** Congestion in the intermediate routers can “hurt” the IATs, making a periodic signal look random.
2. **System Latency:** The capturing NIC (Network Interface Card) might introduce timestamp inaccuracies if the CPU is overloaded.
3. **Clock Drift:** When merging PCAPs from two different capture points, timestamps must be synchronized using NTP/PTP.

Why Time Matters for Encrypted Traffic

Even if a packet is fully encrypted (TLS 1.3), the **timing** cannot be easily hidden without adding significant overhead (padding/delaying).

- **The "Wait" Pattern:** The time between a Client Request and a Server Response (RTT - Round Trip Time) can reveal the processing load of the server, helping AI distinguish between a static webpage and a dynamic database query.

Case Study: Encrypted Traffic Classification

The Problem: How do you identify "Netflix" if the content is hidden inside a TLS tunnel?

Solution: Statistical Fingerprinting

- **Dataset:** VPN-nonVPN dataset (ISCX).
- **Technique:** Use a **Random Forest** classifier trained on "Time-Series" features.
- **Key Features Used:**
 - *f1*: Maximum packet length in the first 10 packets.
 - *f2*: Standard deviation of inter-arrival times.
 - *f3*: Ratio of bytes sent vs. received.
- **Outcome:** Researchers have achieved over **94% accuracy** in distinguishing between VoIP, File Transfer, and Streaming even within encrypted VPN tunnels.

Challenges and Limitations

- **Concept Drift:** Network protocols evolve (e.g., HTTP/2 to HTTP/3).
 - A model trained today might be obsolete in six months.
- **Dataset Imbalance:** Normal web browsing traffic outnumbers malicious or specialized traffic (e.g., SSH) 1000:1, leading to biased models.
- **Real-time Constraints:** In a 100Gbps link, the AI must classify packets in microseconds.
- **Labeling Scarcity:** Obtaining massive amounts of labeled data is difficult and expensive.

Feature Selection - Optimizing the Input

- ▶ In modern network traffic classification, a single flow can generate over **200+ features** (statistical, temporal, header-based).
- ▶ Using all of the features is often counterproductive.
- ▶ **Feature Selection** is the process of selecting the most "informative" subset of features to build the model.

Why Filtering the Features?

- ▶ **The "Curse of Dimensionality":** Too many features can lead to Overfitting, where the AI memorizes the training data noise instead of learning general traffic patterns.
- ▶ **Computational Efficiency:** In high-speed networks, calculating 200 features per flow is too slow. Reducing this to 10-15 key features allows for Line-Rate Classification.
- ▶ **Model Interpretability:** It is easier for network engineers to trust an AI if they know it is making decisions based on specific metrics (like Standard Deviation of Packet Length) rather than obscure variables.

Measuring Feature Importance (Information Gain)

In networking, we often use **Information Gain** to rank features. A feature with high "Information Gain" is one that clearly separates two classes.

- **Low Gain:** *IP Identification Field* (mostly random, doesn't help identify "YouTube").
- **High Gain:** *Maximum Packet Length* (clearly separates "Small-packet DNS" from "Large-packet Video").

Redundancy vs. Relevance

- ▶ A common mistake in network AI is keeping features that are relevant but redundant.
- ▶ **Example:** Forward Mean Packet Length and Forward Max Packet Length.
- ▶ Both are highly relevant for identifying streaming, but because they move together (highly correlated), keeping both increases the model's memory footprint without adding new "knowledge."

Top "Golden" Features in Traffic AI

Recent research studies consistently show that these 5 features often provide 80% of the classification power:

1. **Standard Deviation of Packet Length** (Captures burstiness).
2. **Mean Forward Inter-Arrival Time** (Captures the "rhythm").
3. **Initial TCP Window Size** (Unique to different Operating Systems).
4. **Total Backward Bytes** (Identifies "Download" heavy flows).
5. **Payload-to-Header Ratio** (Distinguishes between data-rich apps and signaling apps).