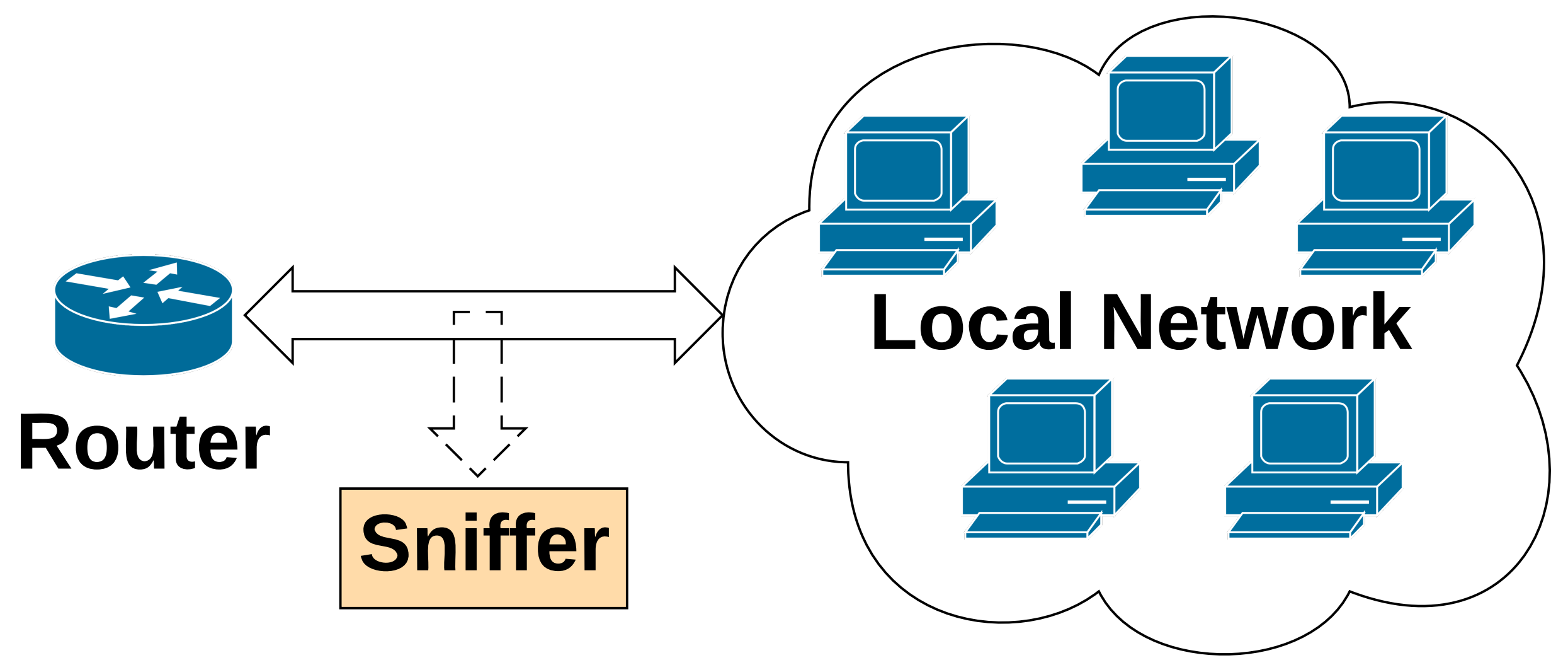
# Machine Learning for Network Intrusion Detection

Luke Hsiao (lwhsiao@stanford.edu) & Stephen Ibanez (sibanez@stanford.edu)

Stanford University

## Introduction and Background

In recent years, networks have become an increasingly valuable target of malicious attacks due to the increased amount of user data they contain. In defense, Network Intrusion Detection Systems (NIDSs) have been developed to detect and report suspicious activity (i.e. an attack). In this project, we explore unsupervised learning techniques for building NIDs, which only analyze unencrypted packet header fields and can run online.



### Design Goals

- Online system that only uses unencrypted packet header fields as features
- Maximize F1 Score while maintaining under 10 false alarms/day on average.

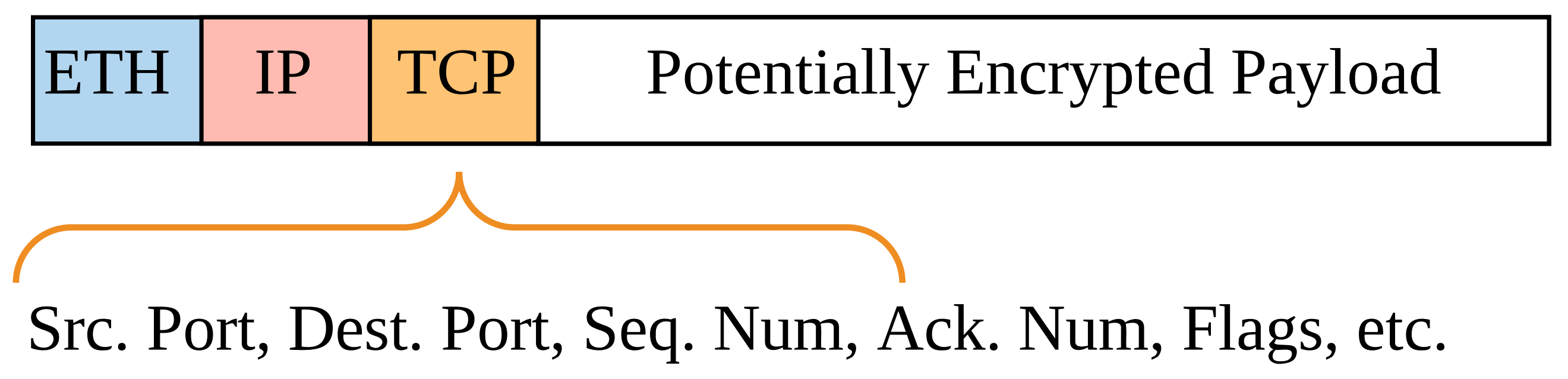
## Dataset & Features

### Dataset

We use the 1999 DARPA Dataset, which contains 5 weeks of `tcpdump` data (including attack free data) collected by a sniffer from simulated network traffic. Weeks 4 and 5 are used as a test set, and contain 201 labeled attack instances interspersed with normal data.

### Features

We use 33 raw unencrypted headers to maintain user privacy.


ETH
IP
TCP
Potentially Encrypted Payload
Src. Port, Dest. Port, Seq. Num, Ack. Num, Flags, etc.

## Models

**Input**: Packets captured via sniffing (e.g. `tcpdump`)

**Output**: Score indicating confidence that packet is anomalous

### Stationary Model: Mixture of Gaussians

**Training** We used a standard mixture of Gaussians model with 16 components and full covariance matrices in order to best fit potential trends. Because we are using raw values of header fields as our features, we use standard scaling techniques to bring the data into a standard range.

**Detection** Whereas mixture of Gaussians can be used to label data based on which Gaussian is the most probable, since we are performing anomaly detection, we instead look at how unlikely a particular packet's features are given the mixture of Gaussians. Specifically, each packet is assigned 16 probabilities $p$ for being drawn from each of the Gaussians. We score a packet as $1 - \max p$. Packets with a low probability of coming from any of the Gaussians have the highest scores.

### Non-Stationary Model: PHAD-C32

**Training** We learn an estimate of the rate of anomalies for the raw values of each feature. If a particular feature is observed $n$ times, and consists of $r$ distinct values, then we approximate the probability that the next observation is anomalous by $r/n$.

**Detection** We define a factor $t$ for each field as the time since the previous anomaly in a particular field. If an anomaly occurred $t$ seconds ago, then the probability that it will occur in the next second is $1/t$. Then, we score each packet as the sum of inverse probability of each of its anomalous fields.

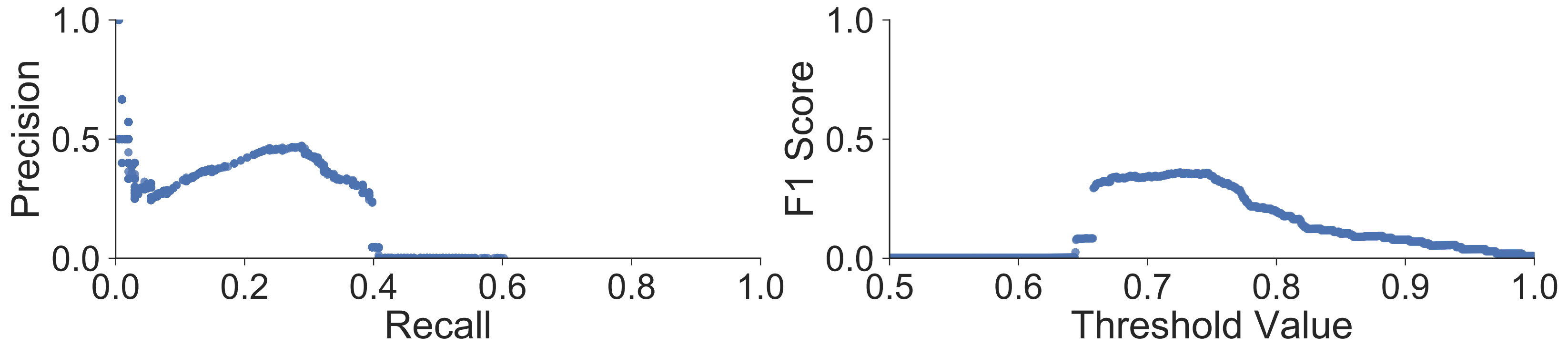
$$\text{score}_{\text{field}} = \frac{t * n}{r}$$

$$\text{score}_{\text{packet}} = \sum_{i \in \text{anomalous fields}} \frac{t_i * n_i}{r_i}$$

**Modeling Anomalies** using raw values results in too many possible values for each field to store in memory. Instead, we follow the clustering algorithm presented in [2] to efficiently approximate whether a value is anomalous.

## Results

**Threshold Tuning** Since the output of both of our models is a score in the interval $[0, 1]$, the threshold at which we classify packets as anomalies allows us to trade-off the recall and precision of our approach. In order to compare with previous methods, we select the threshold which maximizes our F1 score while satisfying our tolerance of false alarms.



**Results** We find that a stationary model such as a mixture of Gaussians, which simply looks at packet headers in isolation, is not able to distinguish between normal packets and attack packets. Incorporating time into a simple non-stationary model achieves results comparable to the top offline systems in the original evaluation [1].

| Approach | Precision | Recall | F1 |
|---|---|---|---|
| Stationary | 2/72 | 2/201 | 0.0147 |
| Non-Stationary | 62/148 | 62/201 | 0.3600 |
| DARPA Expert 1* | 85/185 | 85/201 | 0.4404 |

* This system was an offline system.

## Future Work

- Explore additional features derived from using a stateful NIDS, such as connection information
- Address the shortcomings of the DARPA dataset by mixing real-world data, or create a new, non-simulated dataset.

## Code & References

Check out our code on GitHub:
`github.com/lukehsiao/ml-ids`

[1] R. Lippmann et al. "The 1999 DARPA off-line intrusion detection evaluation". In: *Computer networks* 34.4 (2000), pp. 579–595.

[2] M. V. Mahoney and P. K. Chan. *PHAD: Packet header anomaly detection for identifying hostile network traffic*. Tech. rep. 2001.