

Encrypted Network Traffic Classification

Identifying seven applications from the shape of their encrypted traffic

Sahil Bhatti · M.Tech in Computer Science and Engineering (Data Science), IIT Jammu

Supervisors: Dr. Manoj Singh Gaur (IIT Jammu), Dr. Manjunath Joshi (DA-IICT Gandhinagar)

Submitted September 2022 · Follows the revised edition (October 2026); all numbers are from the approved 2022 thesis

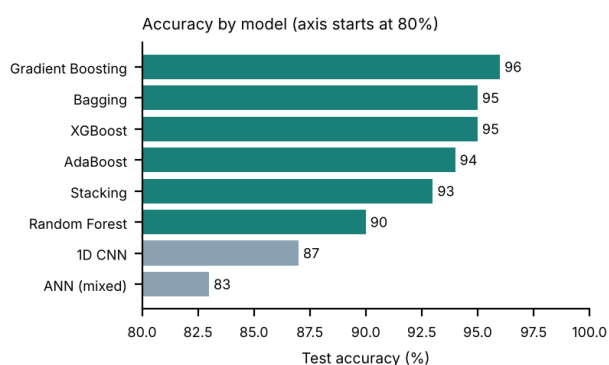
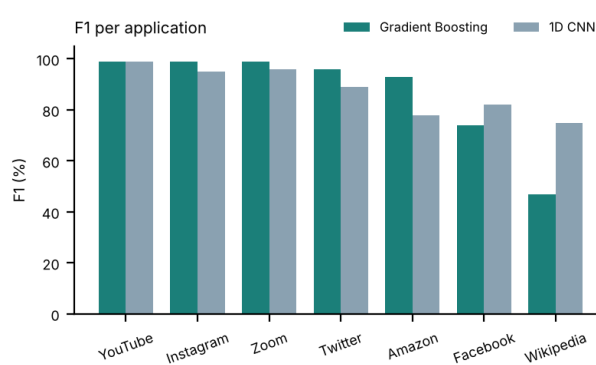
Question

Most web traffic is encrypted, so a network operator can no longer read packets to see which application sent them. What stays visible is the outside of the traffic: packet sizes, timing and direction. **How well can an application be identified from these flow statistics alone?**

Data and method

- Captured my own labelled traffic on the IIT Jammu campus network: Amazon, Facebook, Instagram, Twitter, Wikipedia, YouTube and Zoom, each used on its own for about 24 hours in an Ubuntu VM (Wireshark).
- Converted captures into **18,490 flows** × **81 features** with CICFlowMeter.
- Trained **12 classifiers** on the same features: five fully connected networks (ReLU, Leaky ReLU, ELU, PReLU, mixed), one 1D CNN and six tree ensembles (Bagging, Random Forest, Stacking, AdaBoost, Gradient Boosting, XGBoost).
- Compared models overall and class by class (precision, recall, F1).

Results



- **Tree ensembles worked best.** Gradient Boosting reached 96% accuracy and 96% weighted F1; Bagging and XGBoost 95%. The 1D CNN reached 87%.
- **YouTube, Instagram and Zoom** were recognised almost perfectly by every model; their traffic has a distinct shape (one-sided streaming, two-sided calls, media bursts).
- **Wikipedia and Facebook stayed weak everywhere.** Their flows look like ordinary browsing. Wikipedia precision never rose above 68%.
- A high weighted score can sit on top of a class that fails most of the time, so per-class results must always be reported.

Limits, stated openly

- IP addresses and ports were kept as features, so models may partly learn server addresses; reported accuracy is likely an upper bound for behaviour alone.
- Flows were split at random, not every model was evaluated on the same test set, and each model was run once.
- One machine, one browser, one network, seven applications.

What I would do next

- Rerun cleanly: remove address features, split by time, one fixed test set, repeated seeds with mean and spread.
- Test robustness to evasion: padding and timing changes that make one application imitate another.
- Evaluate under TLS 1.3 and QUIC, and add detection of applications the model has never seen.

The full revised edition (42 pages), with every table and figure and a list of corrections to the 2022 document, is available at bysahil.dev or on request.